ExLibris
RapidILL

Rapid #: -24405268

CROSS REF ID:	**278420**
LENDER:	**QCL (University of Glasgow) :: Main Library**
BORROWER:	**BUF (University at Buffalo, SUNY) :: Main Library**
TYPE:	Article CC:CCG
JOURNAL TITLE:	European journal of mechanics. B, Fluids
USER JOURNAL TITLE:	European Journal of Mechanics - B/Fluids
ARTICLE TITLE:	A stable interface-preserving reinitialization equation for conservative level set method
ARTICLE AUTHOR:	Parameswaran, Parameswaran S.
VOLUME:	98
ISSUE:	
MONTH:	
YEAR:	2023
PAGES:	40-63
ISSN:	0997-7546
OCLC #:	19538423

Processed by RapidX: 4/14/2025 4:41:43 AM

This material may be protected by copyright law (Title 17 U.S. Code)

Contents lists available at ScienceDirect

European Journal of Mechanics / B Fluids

journal homepage: www.elsevier.com/locate/ejmflu

A stable interface-preserving reinitialization equation for conservative level set method

S. Parameswaran, J.C. Mandal [*]

Department of Aerospace Engineering, Indian Institute of Technology - Bombay, Mumbai, Maharashtra, 400076, India

ARTICLE INFO

Article history:
Received 17 July 2022
Received in revised form 28 October 2022
Accepted 1 November 2022
Available online 8 November 2022

Keywords:
Multiphase flow
Finite volume method
Conservative level set method
Stable interface-preserving reinitialization
Reformulation of reinitialization
Artificial compressibility method

ABSTRACT

Existing artificial compression based reinitialization approach for conservative level set method has a few drawbacks, like distortion of fluid–fluid interface and formation of unphysical fluid patches away from the interface. In this paper, a novel reinitialization approach has been presented which circumvents these limitations by reformulating the existing reinitialization equation. The new reinitialization equation by design ensures that the interface contour is kept invariant under the action of level set reinitialization. The unphysical patch formation away from the interface is also resolved here by avoiding the use of ill-conditioned contour normal vectors. Owing to the use of much simpler terms, a significant reduction in the numerical computations is achieved with the new reinitialization equation. The new reinitialization equation also enables one to choose a larger time step during the reinitialization iteration, leading to an overall reduction in computational efforts. In order to evaluate the performance of the present formulation, a set of test problems involving reinitialization of stationary level set functions is carried out first. Then, the efficacy of the proposed reinitialization approach is demonstrated using a set of standard two-dimensional scalar advection based test problems and incompressible two-phase flow problems. Finally, the ability to deal with complex mesh types is demonstrated by solving a test problem on an unstructured mesh consisting of finite volume cells having triangular and quadrilateral shapes.

© 2022 Elsevier Masson SAS. All rights reserved.

1. Introduction

Numerical simulation of incompressible two-phase flows poses great challenges due to the presence of the fluid–fluid interface. Popular contact capturing methods, such as, the Volume of Fluid (VOF) method and the Level Set (LS) method, use an additional interface advection equation along with the incompressible Navier–Stokes (NS) equations. Any inaccuracy in solving the interface advection equation thus significantly affects the quality of the numerical solution. Issues of the formation of jetsam/ floatsam in VOF method, violation of mass conservation in LS method are a few examples associated with errors in the computation of the interface advection equation. In order to overcome the mass conservation error in the classical LS method, a variant of the LS method, known as the Conservative Level Set (CLS) method, is proposed by Olsson and Kreiss [1] and Olsson et al. [2]. The enhanced mass conservation property is achieved here by replacing the signed distance function used in the classical level set method with a hyperbolic tangent type level set function. This level set function is then advected using a scalar conservation law. In order to recover from the excessive numerical dissipation error, an artificial compression based reinitialization procedure is also formulated for the level set function. With the improved mass conservation property, the CLS method shows promising capabilities and provides a good alternative to the classical LS method in solving incompressible two-phase flow problems. However, in practice, the reinitialization procedure for the level set function in CLS method often runs into various numerical difficulties, leading to undesirable results.

Primarily, two major issues with the reinitialization procedure are reported in the literature [3–5]. The first one is the undesired movement of the interface contour during reinitialization. The problem gets further aggravated with the frequent use of reinitialization. This issue arises particularly in two and three dimensions, where, the interface curvature gets involved in the computations. It is demonstrated in Refs. [3,6] that the degree of movement of interface contour depends upon the strength of the interface curvature. Also, a set of numerical experiments presented in Ref. [4] verifies that the frequent use of reinitialization results in a substantial movement of the interface, leading to inaccuracies in the numerical solution. Several attempts to resolve this issue can be found in literature [4,7]. Efforts are focused mainly on localizing the reinitialization process only to a selected region of the level set field. Excessive reinitialization at less

[*] Corresponding author.
E-mail addresses: sanal@aero.iitb.ac.in (S. Parameswaran), mandal@aero.iitb.ac.in (J.C. Mandal).

https://doi.org/10.1016/j.euromechflu.2022.11.001
0997-7546/© 2022 Elsevier Masson SAS. All rights reserved.

dissipated regions is thus avoided. In Ref. [7], the reinitialization is localized by defining a local coefficient based on the degree of sharpness of the level set function. Whereas, in Ref. [4], a metric which depends on the local flow conditions and the numerical diffusion errors is used. It is, therefore, clear that these methods introduce additional complexity and computational efforts in obtaining the local coefficients.

The second issue with the reinitialization procedure is the formation of unphysical fluid patches away from the interface due to the ill-conditioned (high sensitivity towards small changes in the level set function) behaviour of the interface contour normal vectors [8]. The contour normal vectors decide the direction of the compression and diffusion fluxes during the reinitialization process. Far away from the fluid–fluid interface these contour normal vectors become highly sensitive to the changes in the level set function. This results in a noisy vector field, leading to undesirable compression at a far away region and formation of unphysical fluid patches there. Several cures for this problem can be found in literature [3,5,8–11]. In the Accurate Conservative Level Set (ACLS) method, proposed in Ref. [5], the contour normal vectors are computed from an auxiliary signed distance function. The auxiliary signed distance function is constructed here from the level set function using a fast marching method. In the method proposed by Shukla et al. [3], a modified form of the reinitialization equation is used. Here, the level set function is replaced with a smooth function constructed from the level set function itself by employing a mapping procedure. A technique, by combining the reinitialization schemes of both the classical and conservative level set method, named as Improved Conservative Level Set (ICLS) Method, is reported in Ref. [9]. In Ref. [10], a reformulation of the original reinitialization equation which takes care of the spurious movement of interface contour as well as the ill-conditioned behaviour of the contour normal vectors is presented. Later, in Ref. [11], two modifications are proposed for the method given in [10] in order to make it applicable to use while solving the level set advection equation using non-TVD schemes. Recently in Ref. [8], the ill-conditioned unit contour normal vectors are replaced with a different form of normal vectors, such that, the magnitudes of the normal vectors start to diminish away from the interface. This ensures that the reinitialization process is activated only near the interface regions. Though improvements in the contour normal vectors partially circumvent the issue of formation of unphysical fluid patches, they involve solving additional equations adding to the overall computational cost. A brief overview of some of the recent improvements in the artificial compression based reinitialization approach along with their reinitialization equations are provided in Appendix B.

In the present work, a much simpler technique to reinitialize the level set function is presented. Here, the existing artificial compression based reinitialization equation is revised by isolating and removing terms that have potential to move the interface contours. The remaining terms in the modified equation are then reformulated considerably, such that, the use of the contour normal vectors is completely avoided. With the new approach, issues such as the distortion of the interface contour and the unphysical patch formation away from the interface are resolved. In addition, the absence of a viscous dissipation like (second derivative) term in the new reinitialization equation enables one to choose a much larger time step during the reinitialization iteration. The simplified terms also help in significantly reducing the numerical computations per reinitialization time step, aiding an overall reduction in computational efforts. In order to demonstrate the efficacy of the proposed reinitialization scheme, a set of standard two-dimensional test problems involving reinitialization of stationary level set functions (henceforth, we name it as in-place reinitialization problems), advection of level set function under predefined velocity fields and a few standard incompressible two-phase flow problems are solved. Finally, in order to demonstrate the ability to deal with complex mesh types, an incompressible two-phase flow problem is solved on an unstructured mesh consisting of finite volume cells having triangular and quadrilateral shapes.

Rest of the paper is organized as follows. The original CLS method and its reinitialization scheme is briefly described in Section 2. The limitations of the existing artificial compression based reinitialization approach and its new reformulation are also discussed in the same section. In Section 3, the mathematical formulation of incompressible two-phase flows is briefly described. The numerical discretization of the governing equations and the new reinitialization equation are also described in Section 3. Several numerical test problems are solved in Section 4. Finally, the conclusions are given in Section 5.

2. Conservative level set method

The fluid–fluid interface in the conservative level set method is represented in the form of an iso-contour of a hyperbolic tangent type level set function, defined as,

$$\psi(\mathbf{x},t) = \frac{1}{1+\exp\left(\frac{-\phi(\mathbf{x},t)}{\varepsilon}\right)} \equiv \frac{1}{2}\left(\tanh\left(\frac{\phi(\mathbf{x},t)}{2\varepsilon}\right)+1\right) \tag{1}$$

where, $\phi(\mathbf{x},t)$ is the standard signed distance function defined in terms of the minimum distance $d(\mathbf{x},t)$ from the interface, as,

$$\phi(\mathbf{x},t) = \begin{cases} -d(\mathbf{x},t), & \text{inside the first fluid} \\ 0, & \text{at the fluid–fluid interface} \\ +d(\mathbf{x},t), & \text{inside the second fluid} \end{cases}$$

The function ψ takes a value 0 at regions occupied by the first fluid and 1 at the second fluid. Within a thin transition region between the two fluids, ψ varies smoothly from 0 to 1. Width of the transition region is dictated by the parameter ε. The contour corresponds to $\psi(\mathbf{x},0) = 0.5$ represents the actual fluid–fluid interface. The geometric parameters associated with the interface, such as interface normal vector ($\mathbf{n}$) and interface curvature (κ), are obtained from the level set function as,

$$\mathbf{n} = \frac{\nabla\psi}{|\nabla\psi|} \tag{2}$$

$$\kappa = -\nabla\cdot\mathbf{n} \tag{3}$$

The movement of the fluid–fluid interface is achieved by advecting the level set function according to the flow field, as,

$$\frac{\partial\psi}{\partial t} + \nabla\cdot(\mathbf{u}\psi) = 0 \tag{4}$$

where, $\mathbf{u} = u\hat{\imath} + v\hat{\jmath}$, is the divergence-free velocity field.

2.1. Artificial compression based reinitialization procedure

It is well known that the level set function in the conservative level set method suffers from excessive dissipation due to numerical errors [1]. This leads the level set function to deviate from its original hyperbolic tangent type profile. An artificial compression based reinitialization is developed in Ref. [2] in order to re-establish the pre-specified thickness and the profile of the level set function. The discretized level set advection equation together with the reinitialization should satisfy the following three requirements [1]. Firstly, the method should ensure discrete conservation of mass while advecting the level set function. Secondly, the method should not introduce any spurious oscillations.

Finally, the initial properties of the level set function should be maintained throughout the simulation.

The equation for reinitializing the level set function can be written as per [2] as,

$$\frac{\partial \psi}{\partial \tau_r} + C\left(\nabla \cdot (\psi(1-\psi)\mathbf{n}_0) - \nabla \cdot \left(\varepsilon \left(\nabla\psi \cdot \mathbf{n}_0\right)\ \mathbf{n}_0\right)\right) = 0 \tag{5}$$

where, $\mathbf{n}_0 = \frac{\nabla\psi_0}{|\nabla\psi_0|}$ is the interface contour normal vector defined before the reinitialization starts. The variable τ_r [s] is a time like variable and ψ_0 [no units] is the level set function defined at $\tau_r = 0$. The parameter ε [m] is the width of the transition region and C [m/s] is a kinematic parameter introduced in order to take care of the dimensional correctness. The parameter C is usually taken as unity. In Eq. (5), the second term consists of a compression term and a diffusion term, which balances each other once Eq. (5) converges in time τ_r.

In practice, the reinitialization formulation suffers from various deficiencies. As described in the introduction, the above reinitialization procedure may involve error in the computation of $\mathbf{n}_0$. It may also be noted, the reinitialization equation moves the interface contour based on the local curvature. In order to identify terms that have potential to move the interface contour during the reinitialization, Eq. (5) is rewritten in non-conservative form by expanding the compressive and diffusive terms. The basic idea here is to identify and isolate velocity like terms in Eq. (5). To start with, the identity shown in [10] can be obtained by differentiating Eq. (1) in space, as,

$$\nabla\psi = \frac{\psi(1-\psi)}{\varepsilon}\nabla\phi \tag{6}$$

Taking dot product on both sides of Eq. (6) with $\nabla\phi$ and rearranging terms, the following identity can be obtained, as,

$$\psi(1-\psi) = \left(\frac{\varepsilon\nabla\phi}{|\nabla\phi|^2}\right)\cdot\nabla\psi \tag{7}$$

Now, using the definition $\kappa_0 = -\nabla\cdot\mathbf{n}_0$ and Eq. (7), the compression term in Eq. (5) can be expanded as,

$$\nabla\cdot(\psi(1-\psi)\mathbf{n}_0) = -\left(\frac{\varepsilon\kappa_0\nabla\phi}{|\nabla\phi|^2}\right)\cdot\nabla\psi + (1-2\psi)\nabla\psi\cdot\mathbf{n}_0 \tag{8}$$

Similarly, realizing that $(\nabla\psi\cdot\nabla\mathbf{n}_0)\cdot\mathbf{n}_0 = 0$, for any $\mathbf{n}_0$ such that $|\mathbf{n}_0|$ is constant throughout the domain, the diffusion term in Eq. (5) can also be expanded as,

$$\nabla\cdot(\varepsilon(\nabla\psi\cdot\mathbf{n}_0)\mathbf{n}_0) = (-\varepsilon\kappa_0\mathbf{n}_0)\cdot\nabla\psi + \varepsilon(\mathbf{n}_0\cdot\nabla(\nabla\psi))\cdot\mathbf{n}_0 \tag{9}$$

With the above expansions for the compression and diffusion terms, the reinitialization equation (5) can be rewritten as,

$$\frac{\partial\psi}{\partial\tau_r} + \mathbf{v}\cdot\nabla\psi = -C(1-2\psi)\nabla\psi\cdot\mathbf{n}_0 + C\varepsilon\left(\mathbf{n}_0\cdot\nabla(\nabla\psi)\right)\cdot\mathbf{n}_0 \tag{10}$$

where, $\mathbf{v} = \varepsilon\kappa_0 C\left(\mathbf{n}_0 - \frac{\nabla\phi}{|\nabla\phi|^2}\right)$ is a curvature dependent velocity term. Looking at the above form, it is clear that the left hand side (LHS) part of Eq. (10) has the potential to move the interface during the reinitialization iterations according to the velocity vector $\mathbf{v}$. In one dimensional case, κ_0 being zero, there will be no advection of ψ. However, in higher dimensions, κ_0 need not be always zero. A non-zero value of κ_0 will thus result in advection of ψ according to the sign and strength of κ_0 in the direction of $\mathbf{v}$. This will lead to the unwanted movement of interface contour in curved interface regions. In most of the cases, the interface contour movement due to the curvature dependent velocity will be very small. However, when numerical simulations are carried out with a frequent or/and large number of reinitialization iterations, this error starts getting accumulated and the interface movement becomes significant. In order to demonstrate this a set of in-place reinitialization test problems is described in Section 4.1, where, the movement of interface contour is carefully inspected during reinitialization of level set functions with no externally specified velocity field.

The terms in the right hand side (RHS) of Eq. (10) are responsible for the sharpening and dissipating the level set function profile along the contour normal direction. In order to demonstrate this, Eq. (10) is re-written with an appropriate coordinate transformation from the $x-y$ coordinates frame to the contour normal and tangent coordinates frame, $(n_0 - t_0)$, as,

$$\frac{\partial\psi}{\partial\tau_r} + \mathbf{v}_n\cdot\nabla_n\psi = -C(1-2\psi)\frac{\partial\psi}{\partial n_0} + C\varepsilon\frac{\partial^2\psi}{\partial n_0^2} \tag{11}$$

where, $\mathbf{v}_n = \varepsilon\kappa_0 C\left(\hat{n}_0 - \frac{\nabla_n\phi}{|\nabla_n\phi|^2}\right)$ is the curvature dependent velocity term, $\hat{n}_0 = (1,0)$ is the unit vector directed along the contour normal direction and $\nabla_n = \left(\frac{\partial}{\partial n_0}\right)\hat{e}_{n_0} + \left(\frac{\partial}{\partial t_0}\right)\hat{e}_{t_0}$ is the gradient operator in the $n_0 - t_0$ coordinates frame. The level set function, ψ, increases along the contour normal direction. Therefore, the $\frac{\partial\psi}{\partial n_0}$ is always positive. The kinematic constant, C, is taken as unity. Therefore, in the absence of the curvature dependent advection velocity and the second term of RHS in Eq. (11), it is clear that within the interface region, the sign of $\frac{\partial\psi}{\partial\tau_r}$ depends upon the sign of the factor $-(1-2\psi)$. That is, $\frac{\partial\psi}{\partial\tau_r}$ is positive when $\psi > 0.5$ and negative when $\psi < 0.5$. Far away from the interface $\frac{\partial\psi}{\partial n_0}$ is zero, leading to $\dfrac{\partial\psi}{\partial\tau_r} = 0$. This term, therefore, is responsible for sharpening the level set profile along the normal direction during reinitialization. The second term in the RHS of Eq. (11) is a second derivative term, and is responsible for the dissipation of the ψ-profile in the contour normal direction, thus, balancing the interface sharpening term.

2.2. Reformulation of the reinitialization equation

As described in the previous section, the first and second terms in the RHS of Eq. (10) are needed for sharpening the level set profile and balancing it. However, the curvature dependent advection term, $\mathbf{v}\cdot\nabla\psi$, unnecessarily moves the interface; thus it is undesired in a reinitialization procedure. Therefore, in the new formulation of the reinitialization equation, we remove the curvature dependent advection term from Eq. (10). It may be noted, the computation of the level set sharpening and balancing terms as given in the RHS of Eq. (10) are unsuitable in their present form due to the increased sensitive towards numerical errors arising from the ill-conditioned behaviour of the contour normal vectors. In order to overcome the difficulty to deal with these terms, in the new formulation, we seek for terms that are easy to compute and are less sensitive to numerical errors arising form the ill-conditioned contour normal vectors. By taking dot product on both sides of Eq. (6) with $\mathbf{n}_0$, we can get the following identity, as,

$$\nabla\psi\cdot\mathbf{n}_0 = \frac{\psi(1-\psi)}{\varepsilon}\nabla\phi\cdot\mathbf{n}_0 \tag{12}$$

Now, considering $\mathbf{n}_0 \sim \frac{\nabla\phi}{|\nabla\phi|}$ and $|\nabla\phi| \simeq 1$ within the thin interface region, the following simplified relationship can be obtained,

$$\nabla\psi\cdot\mathbf{n}_0 \sim \frac{1}{\varepsilon}\psi(1-\psi) \tag{13}$$

Similarly, in order to obtain a simplified expression for the second term in the RHS of Eq. (10), one can write,

$$(\mathbf{n}_0\cdot\nabla(\nabla\psi))\cdot\mathbf{n}_0 = \nabla(\nabla\psi\cdot\mathbf{n}_0)\cdot\mathbf{n}_0 - (\mathbf{n}_0\cdot\nabla\mathbf{n}_0)\cdot\nabla\psi \tag{14}$$

Due to the fact that $|\mathbf{n}_0| = 1$, the second term in the RHS of Eq. (14) is zero. Now, replacing the $(\nabla\psi\cdot\mathbf{n}_0)$ from the first term using the relation given in (13) and by considering $\mathbf{n}_0 \sim \frac{\nabla\psi}{|\nabla\psi|}$, the following simplified relationship can be obtained,

$$(\mathbf{n}_0\cdot\nabla(\nabla\psi))\cdot\mathbf{n}_0 \sim \frac{1}{\varepsilon}(1-2\psi)|\nabla\psi| \tag{15}$$

It may be evident that the relationships (13) and (15) give alternative approximate terms that are much simpler than the original ones. Particularly, they do not have contour normal vectors. It is demonstrated later in this paper that with the alternative expressions, the new reinitialization equation reinitializes the level set function much more efficiently. After removing the curvature dependent velocity term and by using the simplified expressions given by (13) and (15), Eq. (10) can be reformulated as,

$$\frac{\partial\psi}{\partial\tau_r} = \frac{C}{\varepsilon}\Big(-\psi(1-\psi)(1-2\psi)+\varepsilon(1-2\psi)|\nabla\psi|\Big) \tag{16}$$

Now, defining a new pseudo time variable, as, $\tau_n = \frac{C}{\theta\varepsilon}\tau_r$. Here, the parameter θ [1/s] is a constant and introduced here only to take care of the dimensional correctness. The parameter θ can therefore be taken as unity. By using the rule for the change of variable from τ_r to τ_n in the above equation, the final reformulated reinitialization equation can be written as,

$$\frac{\partial\psi}{\partial\tau_n} = \theta\left[-\psi(1-\psi)(1-2\psi)+\varepsilon(1-2\psi)|\nabla\psi|\right] \tag{17}$$

In order to prove that the hyperbolic tangent function, as given in Eq. (1), trivially satisfies the steady state form of the new reinitialization equation (17), let us consider that the steady state solution of Eq. (17) takes the following form similar to Eq. (1), as,

$$\psi = \frac{1}{1+e^{-\Phi/\varepsilon}} \tag{18}$$

Notice that, here, the scalar field Φ need not necessarily be defined as a signed distance function. Rather, it can be any real-valued function. However, we prove that Eq. (18) satisfies the steady state form of Eq. (17) if and only if the scalar field Φ is a signed distance function. In order to show this, we first take the gradient of Eq. (18), as,

$$\nabla\psi = \frac{e^{\Phi/\varepsilon}}{\varepsilon(1+e^{\Phi/\varepsilon})^2}\nabla\Phi \tag{19}$$

Since the exponential function is always positive, one may also say that,

$$\varepsilon|\nabla\psi| = \frac{e^{\Phi/\varepsilon}}{(1+e^{\Phi/\varepsilon})^2}|\nabla\Phi| \tag{20}$$

Similarly, we may obtain the term $\psi(1-\psi)$ directly from Eq. (18) as,

$$\psi(1-\psi) = \frac{e^{\Phi/\varepsilon}}{(1+e^{\Phi/\varepsilon})^2} \tag{21}$$

Now, one can see that at the steady state, Eq. (17) simplifies to the following relation:

$$-\psi(1-\psi)+\varepsilon|\nabla\psi| = 0 \tag{22}$$

Upon substituting Eqs. ((20) and (21)) in the steady state form of the reinitialization equation, we arrive at the condition that Eq. (22) will be satisfied if and only if $|\nabla\Phi| = 1$. In other words, the scalar-valued function Φ, used in Eq. (18), must be a signed distance function.

The overall behaviour of Eq. (17) may also be described by considering each term in the RHS individually. In the absence of the second term in RHS, Eq. (17) behaves like an ordinary differential equation with $\psi = 1$ and $\psi = 0$ as two stable equilibrium points and $\psi = 0.5$ as an unstable equilibrium point. Fig. 1 shows the phase plot of Eq. (17) with the only first term in the RHS. From the phase plot, it is clear that the first term results in sharpening the level set function profile. Nature of the second term in the RHS of Eq. (17) is to balance the first term. Since the ε and $|\nabla\psi|$ are always positive quantities, the sign of the second term depends on the sign of $(1-2\psi)$. That is, the sign of the second term is positive when $\psi < 0.5$ and negative when $\psi > 0.5$. At $\psi = 0.5$, the second term is zero. In other words, the second term drives the level set function towards a flat profile with $\psi = 0.5$ everywhere, thus balancing the first term. With the above-mentioned sharpening and balancing actions, Eq. (17) reinitializes the level set function by keeping the $\psi = 0.5$ contour invariant during the reinitialization iterations.

One may also notice that both the terms in the RHS of Eq. (17) becomes negative when $\psi > 1$. Similarly, both the terms become positive when $\psi < 0$. This behaviour helps in stabilizing the overshoot ($\psi > 1$) and undershoot ($\psi < 0$) issues arising in case of the use of non-TVD numerical schemes for solving the level set advection equation.

Unlike the artificial compression based reinitialization equation, Eq. (17) is not written in a conservation law form. Therefore, it is not straightforward to strictly prove the integral conservation of the ψ-field over the entire computational domain. However, owing to the interface-preserving property, the area occupied by the interface contour will be unchanged during the reinitialization. Later, in Section 4, it is demonstrated that the new approach is able to achieve a satisfactory level of overall mass conservation in incompressible two-phase flow problems.

3. Mathematical formulation of incompressible two-phase flows

3.1. Governing equations

A dual time-stepping based artificial compressibility approach is followed here for modelling incompressible two-phase flows. The governing system of equations describing the unsteady incompressible viscous two-phase flow can be written as,

$$\frac{\partial\mathbf{U}}{\partial\tau} + I^t\frac{\partial\mathbf{U}}{\partial t} + \left[\frac{\partial(\mathbf{F}-\mathbf{F_v})}{\partial x} + \frac{\partial(\mathbf{G}-\mathbf{G_v})}{\partial y}\right] = \mathbf{F_g} + \mathbf{F_s} \tag{23}$$

where,

$$\mathbf{U} = \begin{Bmatrix} p/\beta \\ \rho u \\ \rho v \\ \psi \end{Bmatrix};\ \mathbf{F} = \begin{Bmatrix} u \\ \rho u^2+p \\ \rho uv \\ u\psi \end{Bmatrix};\ \mathbf{G} = \begin{Bmatrix} v \\ \rho uv \\ \rho v^2+p \\ v\psi \end{Bmatrix};$$

$$\mathbf{F_v} = \begin{Bmatrix} 0 \\ 2\mu\frac{\partial u}{\partial x} \\ \mu(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}) \\ 0 \end{Bmatrix};\ \mathbf{G_v} = \begin{Bmatrix} 0 \\ \mu(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}) \\ 2\mu\frac{\partial v}{\partial y} \\ 0 \end{Bmatrix};$$

$$\mathbf{F_g} = \begin{Bmatrix} 0 \\ -\rho g_x \\ -\rho g_y \\ 0 \end{Bmatrix};\quad \mathbf{F_s} = \begin{Bmatrix} 0 \\ -\sigma\kappa\frac{\partial\psi}{\partial x} \\ -\sigma\kappa\frac{\partial\psi}{\partial y} \\ 0 \end{Bmatrix};\quad I^t = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

where, the vector $\mathbf{U}$ in Eq. (23) denotes the vector of conservative variables and the vectors $(\mathbf{F},\mathbf{G})$ and $(\mathbf{F_v},\mathbf{G_v})$ denote the convective and viscous flux vectors respectively. The vectors $\mathbf{F_g}$ and $\mathbf{F}_s$ denote the source terms containing gravitational and surface tension forces respectively. Here, the surface tension term is modelled using a continuum surface forces (CSF) method, proposed by Brackbill et al. [12]. The parameter σ denotes the surface

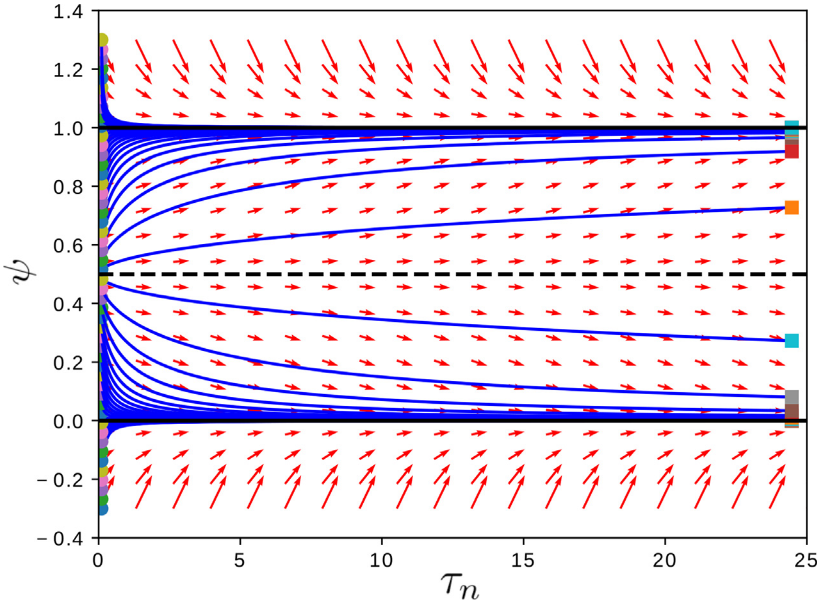


Fig. 1. Phase plot for the new reinitialization equation. The arrows indicate the magnitude and direction of change of ψ with respect to τ_n. The blue lines show the trajectories of different initial values of ψ as τ_n progresses. The initial and final values of ψ are represented using circles and squares respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

tension coefficient per unit length of interface and g_x and g_y denote the x and y components of acceleration due to gravity. variables p, ρ and μ denote the pressure, density and dynamic viscosity of the fluid respectively. The variables τ and t appearing in Eq. (23) denote the pseudo time and the real time respectively. The parameter β denotes the artificial compressibility parameter, which is usually taken as a constant for a given test problem. The artificial compressibility term added in the continuity equation is similar to the one introduced by Chorin in [13]. Once equation (23) converges to a pseudo-steady state, it recovers the set of unsteady incompressible two-phase flow equations. It can be noticed that the level set advection, described by Eq. (4), is combined here with the system of Eqs. (23), and solved simultaneously along with the Navier–Stokes equations. The density and viscosity used in Eq. (23) are defined in terms of the level set function, as,

$$\rho = \rho(\psi) = \rho_2\psi + (1-\psi)\rho_1 \tag{24}$$

$$\mu = \mu(\psi) = \mu_2\psi + (1-\psi)\mu_1 \tag{25}$$

where the subscripts “1” and “2” indicate the properties correspond to the first and the second fluids respectively.

3.2. Numerical discretization of governing equations

A finite volume approach is followed here for solving the governing system of Eqs. (23). In order to proceed with finite volume discretization, the governing system of Eqs. (23) is first integrated over a control volume. The computational domain is then discretized into a finite number of non-overlapping finite volume cells. The final space discretized form of Eqs. (23) for an i^{th} finite volume cell can be written as,

$$\Omega_i \frac{\partial \overline{\mathbf{U}}}{\partial \tau} + I^t \Omega_i \frac{\partial \overline{\mathbf{U}}}{\partial t} + R\left(\overline{\mathbf{U}}\right) = 0 \tag{26}$$

where,

$$R\left(\overline{\mathbf{U}}\right) = \sum_{m=1}^{M}\left(\mathbf{F}\, n_x^m + \mathbf{G}\, n_y^m\right)_m \Gamma_m - \sum_{m=1}^{M}\left(\mathbf{F_v}\, n_x^m + \mathbf{G_v}\, n_y^m\right)_m \Gamma_m - \Omega_i \overline{\mathbf{F}}_g - \Omega_i \overline{\mathbf{F}}_s,$$

Ω_i is the area, Γ_m and $\mathbf{n}^m = (n_x^m, n_y^m)$ are the length and edge normals of the m^{th} edge respectively and M is the total number of edges of the finite volume cell i. The vectors $\overline{\mathbf{U}}$, $\overline{\mathbf{F}}_g$ and $\overline{\mathbf{F}}_s$ represent the cell averaged values of $\mathbf{U}$, $\mathbf{F_g}$ and $\mathbf{F_s}$ respectively. The source term vector, $\overline{\mathbf{F}}_g$, appearing in Eq. (26) is computed by multiplying the cell averaged value of density and the acceleration due to gravity. Evaluation of the surface tension vector, $\overline{\mathbf{F}}_s$, involves the computation of gradient, contour normal and curvature of the level set function. The contour normal and curvature are computed from the gradient of the level set function as per equation (2) and Eq. (3) respectively, where the gradient vector is evaluated using a central least square method as explained in [14]. The convective flux vector $\left(\mathbf{F}\, n_x^m + \mathbf{G}\, n_y^m\right)$ and the viscous flux vector $\left(\mathbf{F_v}\, n_x^m + \mathbf{G_v}\, n_y^m\right)$ in Eq. (26) are computed at the edges of each cell using a Roe-type Riemann solver, developed in [15], and a Green–Gauss integral approach over a Coirier’s diamond path, described in [14], respectively. The real time derivatives appearing in Eq. (26) are computed using a three point implicit backward differencing procedure. Finally, an explicit three stage Strong Stability Preserving Runge–Kutta (SSP-RK) method, described in [16], is used for iterating in pseudo-time. The time step required for the pseudo-time iteration is computed by considering the convective, viscous, gravitational and surface tension effects. For faster convergence, a local time stepping approach is adopted here, in which, each cell is updated using its own $\Delta\tau_i$. The local time step $\Delta\tau_i$ is computed as,

$$\Delta\tau_i = \min(\Delta\tau_i^{\text{conv}}, \Delta\tau_i^{\text{visc}}, \Delta\tau_i^{\text{grav}}, \Delta\tau_i^{\text{surf}}) \tag{27}$$

where, $\Delta\tau_i^{\text{conv}}$, $\Delta\tau_i^{\text{visc}}$, $\Delta\tau_i^{\text{grav}}$ and $\Delta\tau_i^{\text{surf}}$ are the maximum allowed time steps due to convective flux, viscous flux and gravitational and surface tension forces respectively. These time steps are evaluated as,

$$\Delta\tau_i^{\text{conv}} = \frac{\nu\ \Omega_i}{\sum_{m=1}^{M}\left(|u_n| + \sqrt{(u_n)^2 + \frac{\beta}{\rho_i}}\right)_m \Gamma_m};$$

$$\Delta\tau_i^{\text{visc}} = \frac{\nu\ \Omega_i^2}{\left(\frac{8}{3}\right)\frac{\mu_i}{\rho_i}\sum_{m=1}^{M}(\Gamma_m)^2};$$

$$\Delta\tau_i^{\text{grav}} = \nu\sqrt{\frac{h}{|\mathbf{g}|}} \qquad \text{and} \qquad \Delta\tau_i^{\text{surf}} = \nu\sqrt{\frac{(\rho_1+\rho_2)h^3}{4\pi\sigma}}$$

where, ν is the Courant number, h is the average cell size and u_n is the velocity component along the edge normal direction. For stability reasons, the Courant number, ν, is always taken less than unity. The $\Delta\tau_i$ computed using Eq. (27) is further restricted based on the real time step [17], as, $\Delta\tau_i \leq \frac{2}{3}\Delta t$. Detailed descriptions of the numerical methods used for solving incompressible two-phase flows are excluded from here due to brevity reasons. One can refer to [14,15] for more details.

3.3. Numerical discretization of reinitialization equation

The RHS of Eq. (17) consists of two terms. In order to evaluate the first term at a given cell, only the cell centre values of ψ is sufficient. However, evaluation of the second term involves computation of the gradient of ψ. Here, the gradient of ψ at the cell centres are evaluated using a central least square approach as discussed in Appendix A. Finally, the time integration of Eq. (17) is carried out using a three stage Strong Stability Preserving Runge–Kutta (SSP-RK-3) method described in Section 3.3.1.

3.3.1. Time integration for the reinitialization equation

According to the SSP-RK-3 approach described in [16], the cell averaged value of the unknown function in Eq. (17) is updated as,

$$\begin{aligned}
\overline{\psi_i}^{(1)} &= \overline{\psi_i}^{(n)} + \frac{\Delta\tau_n}{\Omega_i}\, L\left(\overline{\psi_i}^{(n)}\right) \\
\overline{\psi_i}^{(2)} &= \frac{3}{4}\overline{\psi_i}^{(n)} + \frac{1}{4}\overline{\psi_i}^{(1)} + \frac{1}{4}\frac{\Delta\tau_n}{\Omega_i}\, L\left(\overline{\psi_i}^{(1)}\right) \\
\overline{\psi_i}^{(n+1)} &= \frac{1}{3}\overline{\psi_i}^{(n)} + \frac{2}{3}\overline{\psi_i}^{(2)} + \frac{2}{3}\frac{\Delta\tau_n}{\Omega_i}\, L\left(\overline{\psi_i}^{(2)}\right)
\end{aligned} \tag{28}$$

where, $\overline{\psi_i}^{(n)}$ and $\overline{\psi_i}^{(n+1)}$ are the cell averaged level set function defined at n^{th} and $(n+1)$th time levels, respectively, $\overline{\psi_i}^{(1)}$ and $\overline{\psi_i}^{(2)}$ are the intermediate values of ψ and

$$L\left(\overline{\psi_i}^{(\cdot)}\right) = \theta\left[-\overline{\psi_i}^{(\cdot)}(1-\overline{\psi_i}^{(\cdot)})(1-2\overline{\psi_i}^{(\cdot)}) + \varepsilon|\nabla\overline{\psi_i}^{(\cdot)}|(1-2\overline{\psi_i}^{(\cdot)})\right] \tag{29}$$

.

For the explicit time integration of Eq. (17), the time step is restricted based on the nature of the reinitialization equation. In order to find out the allowable time step, Eq. (17) is rewritten using the following identities,

$$\psi(1-\psi) = \varepsilon\left(\frac{1}{|\nabla\phi|}\right)\mathbf{n}\cdot\nabla\psi \tag{30}$$

and

$$|\nabla\psi| = \mathbf{n}\cdot\nabla\psi \tag{31}$$

as,

$$\frac{\partial\psi}{\partial\tau_n} + \boldsymbol{\mathcal{S}}\cdot\nabla\psi = 0 \tag{32}$$

where, $\boldsymbol{\mathcal{S}} = \varepsilon\theta(1-2\psi)\left(\frac{1-|\nabla\phi|}{|\nabla\phi|}\right)\mathbf{n}$. Eq. (32) is in Hamilton–Jacobi form with a velocity like variable, $\boldsymbol{\mathcal{S}}$ [m/s], aligned in the direction of the vector field $\mathbf{n}$. Note that, unlike the curvature dependent velocity vector, $\mathbf{v}$, found in Eq. (10), the vector $\boldsymbol{\mathcal{S}}$ has no dependency on the curvature of the fluid–fluid interface. The vector $\boldsymbol{\mathcal{S}}$ has opposite signs on both sides of the interface and is equal to zero at $\psi = 0.5$. According to the sign of the term $\left(\frac{1-|\nabla\phi|}{|\nabla\phi|}\right)$, the vector field $\boldsymbol{\mathcal{S}}$ will be directed either towards or away from the interface, resulting only in sharpening or flattening of the ψ profile and not in any apparent motion of the interface contour. Since the solution variable ψ is updated according to the sharpening velocity vector $\boldsymbol{\mathcal{S}}$, a stable explicit time integration scheme for Eq. (32) is possible only with a restricted time step based on the CFL condition as,

$$\Delta\tau_n \leq \frac{h}{|\boldsymbol{\mathcal{S}}|} \tag{33}$$

where, h is the average cell size and $|\boldsymbol{\mathcal{S}}| = \varepsilon\,\theta\,|(1-2\psi)|\,\left|\left(\frac{1-|\nabla\phi|}{|\nabla\phi|}\right)\right|$. Under normal conditions, the value of ψ is bounded between 0 and 1, which implies, $|1-2\psi|_{\max} = 1$. Similarly, considering $|\nabla\phi| \simeq 1$, the term $\left|\left(\frac{1-|\nabla\phi|}{|\nabla\phi|}\right)\right|$ becomes negligibly small. With the above approximations and by taking $\theta = 1$, one may obtain an upper bound for the magnitude of $\boldsymbol{\mathcal{S}}$, as, $|\boldsymbol{\mathcal{S}}|_{\max} = \varepsilon$. If one choose the thickness parameter, ε, similar to [1], as, $\varepsilon = \frac{h^{(1-d)}}{2}$, the CFL condition given in Eq. (33) yield the allowable time step as,

$$\Delta\tau_n \leq 2h^d \tag{34}$$

It may be noticed that the time step, $\Delta\tau_n$, given in Eq. (34) is larger by a factor of $4/h$ in comparison with the allowable time step for the artificial compression based reinitialization procedure [1]. The presence of a viscous dissipation term in the artificial compression based approach restricts the reinitialization time step to a smaller value [1]. In addition to the larger time step, the relatively less complex terms in the new reinitialization equation significantly reduces the numerical computations per reinitialization time step, aiding an overall reduction in computational efforts.

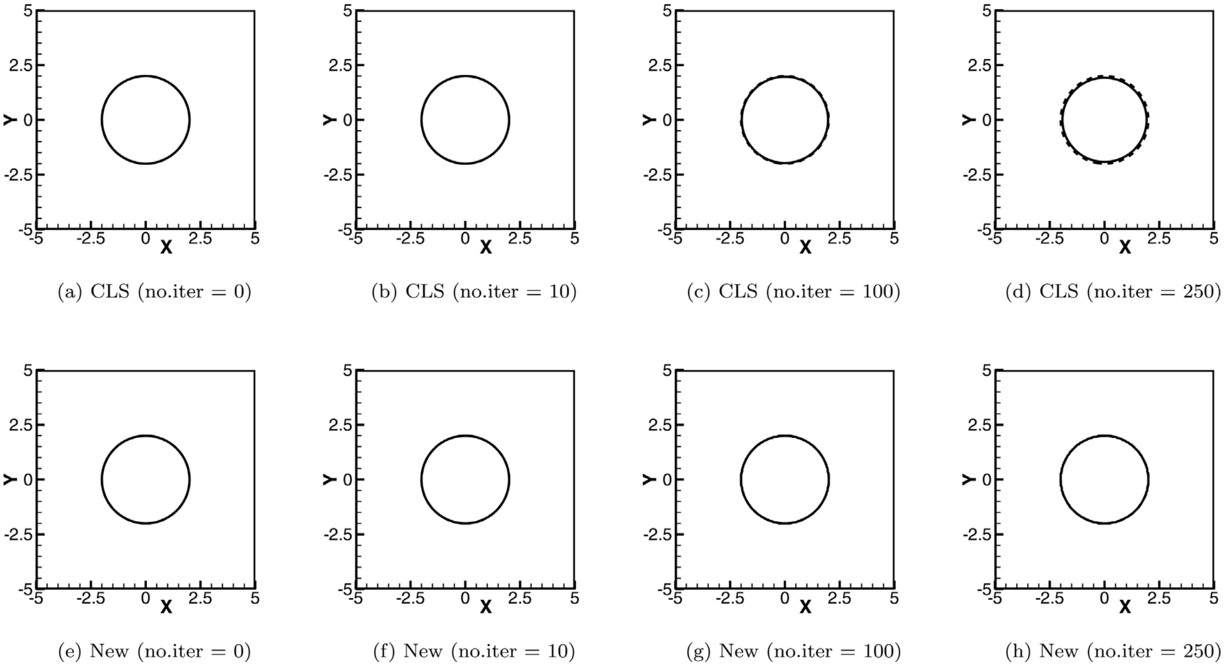
3.3.2. Implementation details of the new reinitialization equation

A summary of the major steps in the numerical implementation of the new reinitialization equation is described here. As soon as the advection step of the level set function is completed, the following reinitialization steps are performed:

- *Step 1:* Compute the maximum possible $\Delta\tau_n$ using Eq. (34).
- *Step 2:* Compute the derivatives of ψ along x and y directions. Use central differencing in case of structured mesh or use the central least square method described in the Appendix A in case of unstructured meshes.
- *Step 3:* Compute the magnitude of the gradients of ψ using the equation $|\nabla\psi| = \sqrt{\left(\frac{\partial\psi}{\partial x}\right)^2 + \left(\frac{\partial\psi}{\partial y}\right)^2}$ for two dimensional problems.
- *Step 4:* Compute the right hand side of the new reinitialization equation using Eq. (29).
- *Step 5:* Update to a new τ_n level using the SSP-RK-3 steps described in Eq. (28).
- *Step 6:* Repeat *Step 1* to *Step 5* until the iteration converges in τ_n.

4. Numerical experiments

Performance of the new reinitialization procedure is evaluated using three types of test problems. In order to illustrate the movement of the interface contour during the reinitialization and to study convergence behaviours, a set of test problems involving reinitialization of stationary level set function, is carried out first in Section 4.1. These problems are named here as in-place reinitialization problems. After the in-place reinitialization problems, a set of scalar advection based test problems are considered in Section 4.2, where, the area and shape errors during the level set



(a) CLS (no.iter = 0) (b) CLS (no.iter = 10) (c) CLS (no.iter = 100) (d) CLS (no.iter = 250)

(e) New (no.iter = 0) (f) New (no.iter = 10) (g) New (no.iter = 100) (h) New (no.iter = 250)

Fig. 2. In-place reinitialization of a circular interface using the original CLS approach (a) to (d) and the new approach (e) to (h). The dashed black curve denotes the initial interface and the solid black curves denote the interfaces at the respective pseudo-time iterations.

advection are quantified. In Sections 4.4–4.6, several incompressible two-phase flow test problems are presented. These problems are arranged according to their increasing levels of complexities, starting from an inviscid flow problem to problems involving viscous and surface tension forces. All test problems are solved on simple Cartesian type meshes. However, in order to demonstrate the ability to deal with complex mesh types, the last test problem is also solved on an unstructured mesh consisting of finite volume cells having triangular and quadrilateral shapes.

4.1. In-place reinitialization problems

As illustrated in Section 2.1, the artificial compression based reinitialization approach often results in moving the interface contour according to the sign and strength of the interface curvature. This also results in poor convergence during the reinitialization iterations. In order to demonstrate this, a set of test problems involving reinitialization of stationary level set function is carried out here. Tests are performed using the new reinitialization approach (denoted as “new”) and the artificial compression based reinitialization procedure of [2] (denoted as “CLS”). Firstly, the problem of movement of interface contours is demonstrated, using three shapes having different curvature profiles, in Section 4.1.1. Secondly, the convergence behaviour of the reinitialization procedure is studied in Section 4.1.2. Finally, in Section 4.1.3, the CPU time for one reinitialization iteration is measured and compared.

4.1.1. Demonstrating movement of interface contours

In order to demonstrate this, level set functions corresponds to three basic geometric shapes of different curvature profiles are constructed first. For the present study, a circle with a diameter of 4 units, an ellipse with 4 units and 2 units of major and minor axes respectively and a square with size 3 units, are chosen. These geometric shapes are placed at the centre of a computational domain of a square shaped region bounded between $-5 \leq x \leq 5$ and $-5 \leq y \leq 5$. The computational domain is discretized using a 200×200 Cartesian mesh. These level set functions are then taken as the initial condition for both the reinitialization equations. In order to see how stable are the interface contours under the repeated action of reinitialization iterations, a large number of reinitialization iterations using both the reinitialization approaches are carried out for all the three shapes. Unlike other test problems, numerical errors associated with the advection of level set function are not present here. Therefore, this test helps in isolating errors associated only with the reinitialization process. In principle, the reinitialization process should not result in the movement of the interface contour. However, due to the curvature dependent velocity field implicitly present in the artificial compression based reinitialization approach, as explained in Section 2.1, one may expect to see a spurious movement of the interface contours in case of the existing artificial compression based approach.

During the reinitialization iterations, the deformation of the interface contour in both the CLS and the new approaches are monitored constantly. Figs. 2, 3 and 4 show the interface contours during the in-place reinitialization compared with the initial contours in cases of circle, ellipse and square shapes respectively. The solid black curve denotes the interface contour during reinitialization and the dashed black curve denotes the initial interface contour. One can notice that, for all the three shapes, up to 10 number of reinitialization iterations no significant changes in the interface contours are visible. However, as the number of reinitialization iterations increases, the interface contours, in case of the CLS approach, show deformations. Especially, more deformations can be observed at regions having higher curvature. Whereas, there are no visible deformations of the interface contours even after 250 iterations in case of the new reinitialization approach. In order to quantify the degree of movement of the interface contours, an L^2 error is defined, as,

$$L^2 = \left(\frac{1}{Nx \times Ny}\right) \sqrt{\sum_{i=1}^{Nx} \sum_{j=1}^{Ny} \left(\psi_{ij}^{l} - \psi_{ij}^{0}\right)^2} \tag{35}$$

where, the superscripts, 0 and l, denote the discretized ψ-field at initial and at l^{th} reinitialization iteration levels respectively, and N_x and N_y denote the number of cells along x and y directions

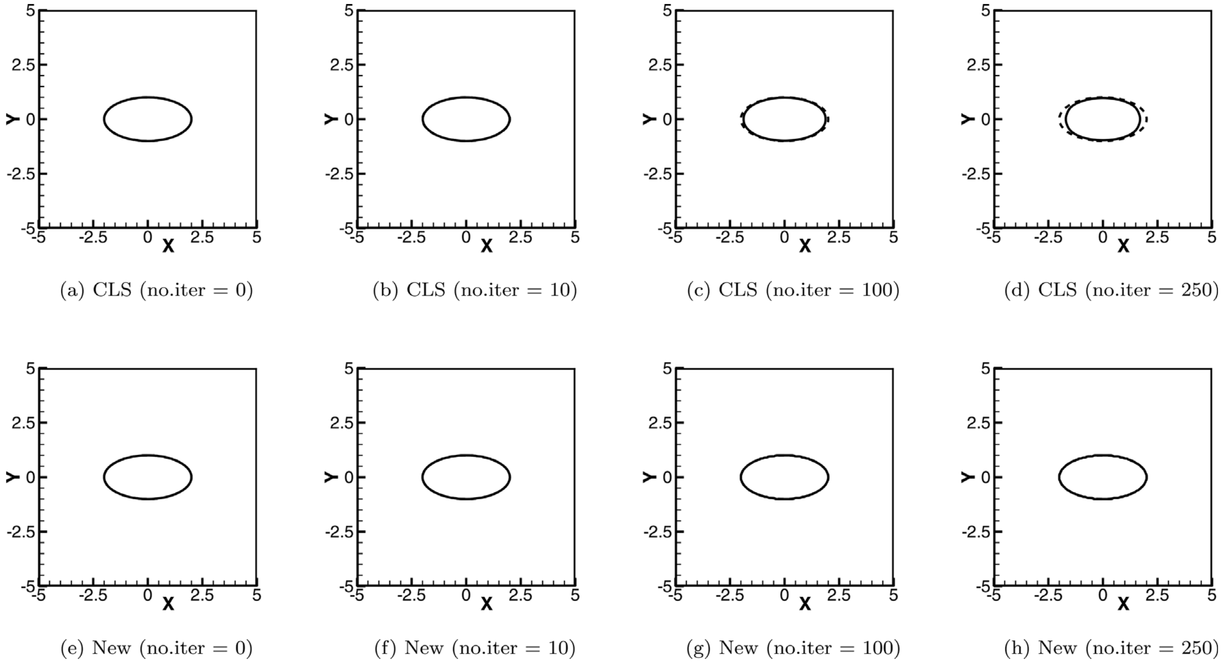


Fig. 3. In-place reinitialization of an elliptical interface using the original CLS approach (a) to (d) and the new approach (e) to (h). The dashed black curve denotes the initial interface and the solid black curves denote the interfaces at the respective pseudo-time iterations.

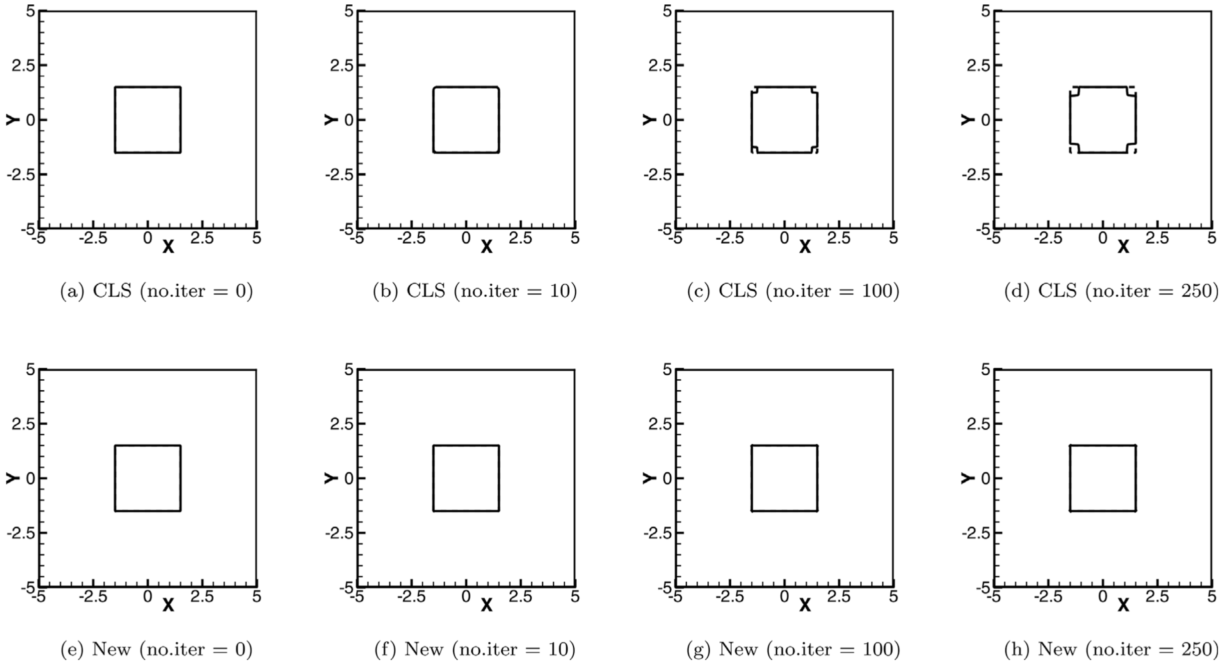
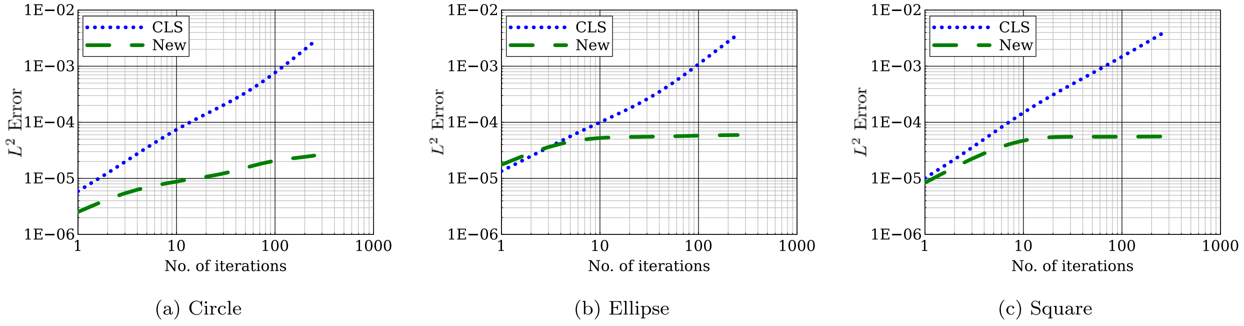


Fig. 4. In-place reinitialization of a square shaped interface using original CLS approach (a) to (d) and the new approach (e) to (h). The dashed black curve denotes the initial interface and the solid black curves denote the interfaces at the respective pseudo-time iterations.

respectively. Fig. 5 shows the variation of the L^2 error with respect to the number of reinitialization iteration. From Fig. 5, one can see that the L^2 error in case of the new reinitialization scheme is below 10^{-4} level. Whereas, in case of CLS, the L^2 error keeps increasing as the number of reinitialization iteration increases. These observations are in well agreement with the discussion given in Sections 2.1 and 2.2.

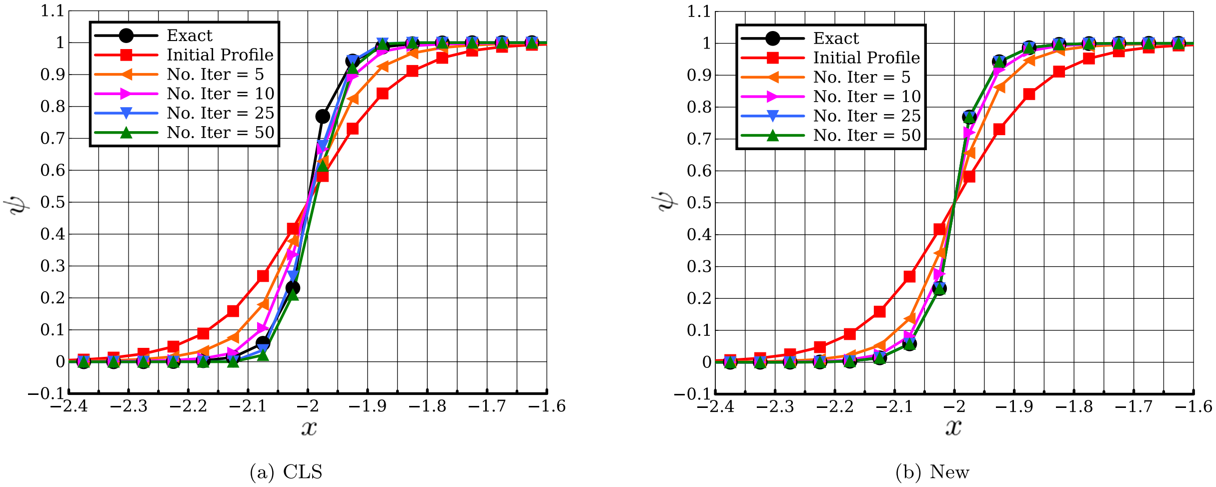
4.1.2. Convergence study of the reinitialization equation

In the previous tests, the issue of the spurious movement of the interface contour for the CLS approach is demonstrated. In this section the convergence behaviours of the two reinitialization approaches are compared. For this, the following test problem is considered. An initial ψ-field, corresponds to a circular interface of a diameter of 4 units, is constructed on a domain bounded between $-5 \leq x \leq 5$ and $-5 \leq y \leq 5$. The ψ-field is constructed in such a way that the initial ψ-profile is diffused over several cells by setting a large $\varepsilon = 2.0\ \Delta x$ in Eq. (1). Now, this diffused ψ-field is used as the initial condition for both the CLS and the new reinitialization equations, where, a smaller $\varepsilon = 0.5\ \Delta x$ is set for the reinitialization equations. During the reinitialization iterations the initially diffused ψ-field is expected



(a) Circle (b) Ellipse (c) Square

Fig. 5. The L^2 error plots with respect to number of reinitialization iteration of the in-place reinitialization of circular, elliptical and square shaped interface solved using CLS and the new reinitialization schemes.



(a) CLS (b) New

Fig. 6. The slice view of a portion of the ψ function along x-axis at different levels of reinitialization iterations of both the CLS and the new reinitialization schemes.

to converge to a sharp ψ-profile corresponding to Eq. (1) with $\varepsilon = 0.5\ \Delta x$. The reinitialization equations are solved on a Cartesian mesh of 200 × 200 cells. In Fig. 6, a slice view of a portion of the ψ function along the x-axis is plotted. From Fig. 6(b) one can see that, with the new reinitialization approach, the initially diffused ψ-profile converges to the exact solution (*ie*., ψ-profile corresponds to Eq. (1) with $\varepsilon = 0.5\ \Delta x$) in around 15 iterations. The level set profile in case of the CLS also gets sharper during the reinitialization iteration. However, the interface contour slowly drifts away from its original position (refer Fig. 6(a)).

In order to quantify the convergence characteristics of the reinitialization equations, the following residual is defined, as,

$$\text{Residual} = \left(\frac{1}{Nx \times Ny}\right)\sqrt{\left(\frac{\psi_{ij}^{l+1} - \psi_{ij}^{l}}{\Delta\tau_p}\right)^2}, \tag{36}$$

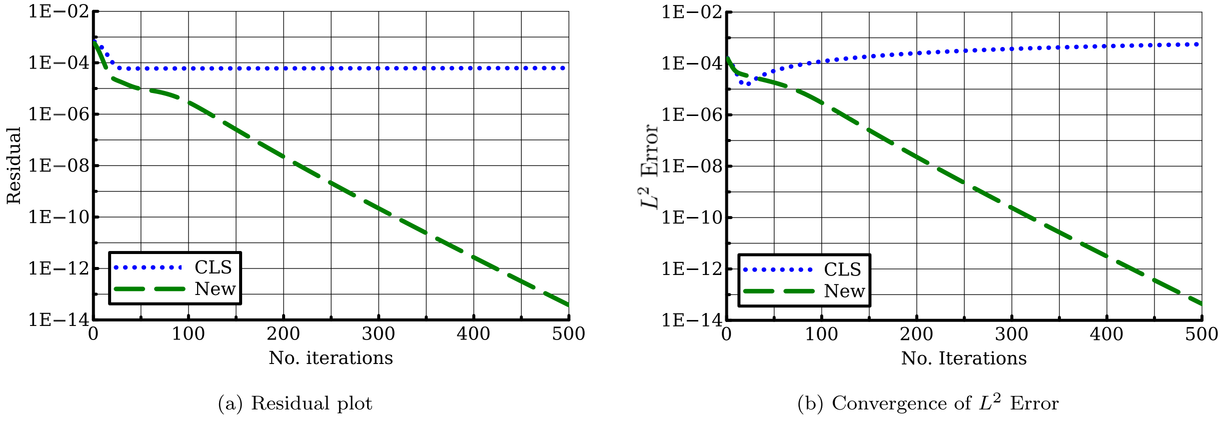
where, τ_p denotes the reinitialization time step ($\tau_p = \tau_r$ in case of CLS reinitialization and $\tau_p = \tau_n$ in case of the new reinitialization equation). An L^2 error is also defined, similar to Eq. (35), where, the ψ_{ij}^0 is replaced with a ψ_{ij}-field which satisfies Eq. (1) with $\varepsilon = 0.5\ \Delta x$. The residual of the ψ-field (Eq. (36)) indicates the changes in the ψ-field during the reinitialization iterations. Whereas, the L^2 error quantifies the convergence of the ψ-profile to the exact ψ-profile. Fig. 7 shows the variation of the residual and the L^2 error with respect to the number of reinitialization iteration. From Fig. 7, one can see that both the residual and the L^2 error of the new reinitialization scheme converges to the machine zero level. Whereas, the residual, in case of the CLS, does not converge below 10^{-5} level. The L^2 error, in case of the CLS continue to increase followed by an initial short dip. The poor convergence behaviour of the CLS could be due to the spurious movement of the interface contour.

4.1.3. Comparison of CPU time per reinitialization iteration

In order to compare the actual CPU time required for one reinitialization iteration, the same test problem is solved on six different meshes ranging from 12 × 12 up to 400×400. Codes are written in standard `C++03` and compiled using `gcc 7.5.0` with `-O0` optimization flag. Tests are carried out under serial computing mode on a linux based operating system with an eighth generation `Intel Core i5` processor. The CPU time required for 500 iteration is measured and the average CPU time for one reinitialization iteration is calculated. Fig. 8 shows the CPU time for one reinitialization iteration for the CLS approach and the new approach. One can see from Fig. 8 that the CPU time for the new reinitialization approach is much less compared to the CLS approach. On average, the new approach is 17.191 times faster than the CLS.

4.2. Reinitialization of scalar advection problems

In order to further study the performance of the new reinitialization formulation, a set of standard two-dimensional scalar advection based test problems are considered next. In the scalar advection problems, the initial interface is placed at (0.25, 0.5) on a unit square domain and advected upon a predefined velocity field. After each scalar advection time step the level set function is reinitialized using a set of four reinitialization iterations. The area confined by the interface and the L^1 and L^2 error norms are



(a) Residual plot

(b) Convergence of L^2 Error

Fig. 7. Residual and the L^2 error plots with respect to number of reinitialization iteration of the CLS and the new reinitialization schemes.

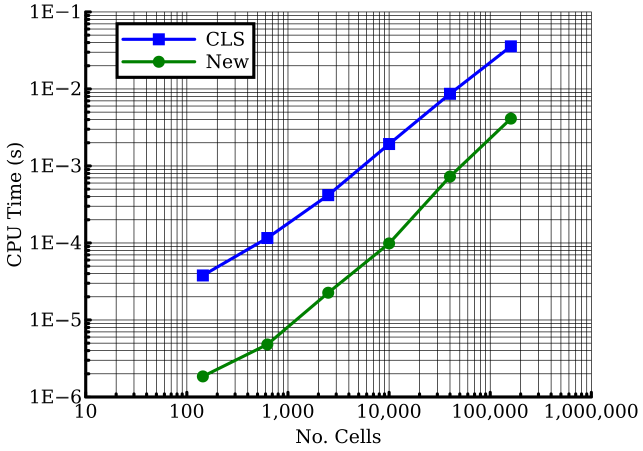


Fig. 8. Average CPU time for one reinitialization iteration for the CLS and the new reinitialization approaches.

monitored during the simulation. The percentage area error, at any given time t, is computed as,

$$\text{Area Error (\%)} = \left(\frac{A^t - A^0}{A^0}\right) \times 100 \tag{37}$$

where, A is the area enclosed by the 0.5 contour of the level set function and the superscript t and 0 represent the data computed at time t and at the initial time, $t = 0$, respectively. Similarly, the L^1 and L^2 error norms at time level $t = 2\pi$, are defined as,

$$L^1 = \left(\frac{1}{Nx \times Ny}\right) \sum_{i=1}^{Nx} \sum_{j=1}^{Ny} |\psi_{ij}^{2\pi} - \psi_{ij}^0| \tag{38}$$

and

$$L^2 = \left(\frac{1}{Nx \times Ny}\right) \sqrt{\sum_{i=1}^{Nx} \sum_{j=1}^{Ny} \left(\psi_{ij}^{2\pi} - \psi_{ij}^0\right)^2} \tag{39}$$

where, $\psi_{ij}^{2\pi}$ and ψ_{ij}^0 are the discretized level set functions defined at time level, $t = 2\pi$ and $t = 0$ respectively.

4.2.1. Reinitialization of circular disc rotation problem

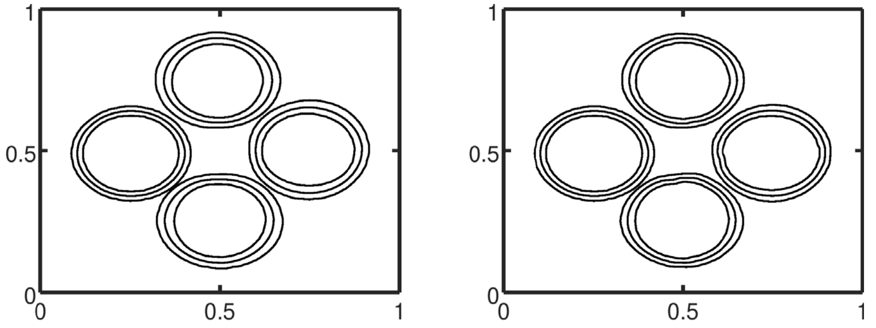
Rotation of a circular disc, similar to the test reported in [1], is considered first, where, a circular disc of radius 0.15 units is advected upon a velocity field $u = (y - 0.5)$ and $v = (0.5 - x)$. The test problem is solved on five levels of Cartesian meshes starting from 25 × 25 up to 400 × 400. The level set contours correspond to $\psi = 0.05, \psi = 0.5$ and $\psi = 0.95$ for the 100 × 100 case at time levels $t = 0$, $t = \pi/4$, $t = \pi/2$ and $t = 3\pi/4$ for the CLS and the new reinitialization approaches are plotted in Fig. 9. The Table 1 shows the percentage area error and error norms for all five mesh levels for the CLS and the new reinitialization approaches. Further, the error norms are plotted against the mesh size in Fig. 10 along with reference slopes for the first and second order rate of convergence. In Fig. 11, the area errors for each mesh levels are plotted with respect to time. Looking at the above figures and table, one can see that the new reinitialization approach shows better performance as compared to the CLS approach.

4.2.2. Reinitialization of Zalesak's disc rotation problem

The second problem considered is the advection of Zalesak's disc, similar to the test reported in [5]. The Zalesak's disc is of radius 0.15 units, and notch length and width of 0.3 units and 0.1 units respectively. Due to the presence of sharp corners, this test problem shows more numerical errors compared to the previous problem. Similar to the rotation of the circular disc problem, this problem also is solved on five levels of Cartesian meshes starting from 25 × 25 up to 400 × 400. The level set contours correspond to $\psi = 0.05, \psi = 0.5$ and $\psi = 0.95$ for the 100 × 100 case at time levels $t = 0$, $t = \pi/4$, $t = \pi/2$ and $t = 3\pi/4$ for the CLS and the new reinitialization approaches are plotted in Fig. 12. In order to compare the performance of the CLS and the new reinitialization approaches quantitatively, the percentage area error and error norms are computed after the disc completed one full rotation (*ie*., at $t = 2\pi$). The Table 2 shows the percentage area error and error norms for all five mesh levels. Further, the error norms are plotted against the mesh size in Fig. 13 along with reference slopes for the first and second order rate of convergence. In Fig. 14, the area errors for each mesh levels are plotted with respect to time. Similar to the previous problem, here also it can be seen that the new reinitialization approach shows better performance as compared to the CLS approach.

4.2.3. Reinitialization of circular disc deformation problem

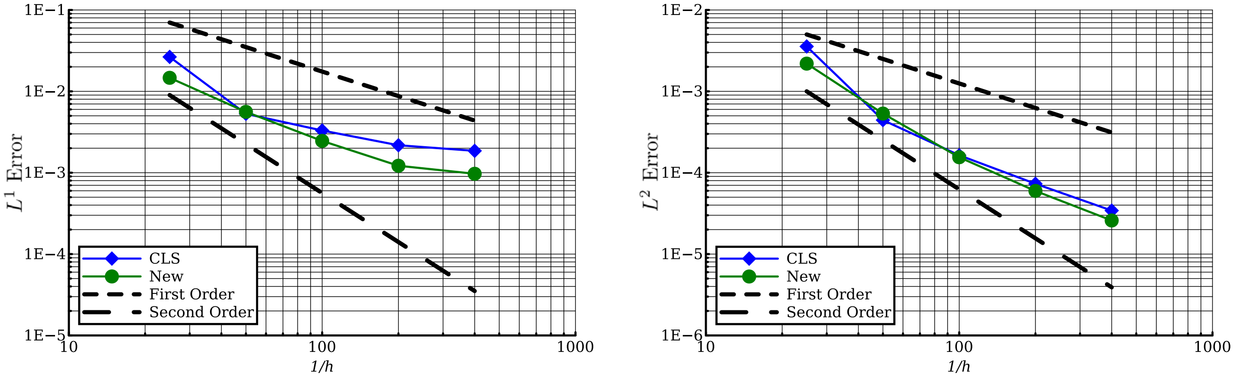
The last scalar advection test problem considered is the advection of a circular disc subjected to a shear velocity field, $u = \sin^2(\pi x)\sin(2\pi y)$, $v = -\sin^2(\pi y)\sin(2\pi x)$. Fig. 15 shows a qualitative comparison of the interface contour for the CLS and the new reinitialization approaches at $t = 4$ s solved a 200 × 200 Cartesian mesh. From Fig. 15, one can clearly see that the tail of the interface is fully resolved without breaking in case of the new reinitialization scheme. By strictly ensuring that the interface contours do not move during the level set reinitialization, the new



(a) CLS

(b) New

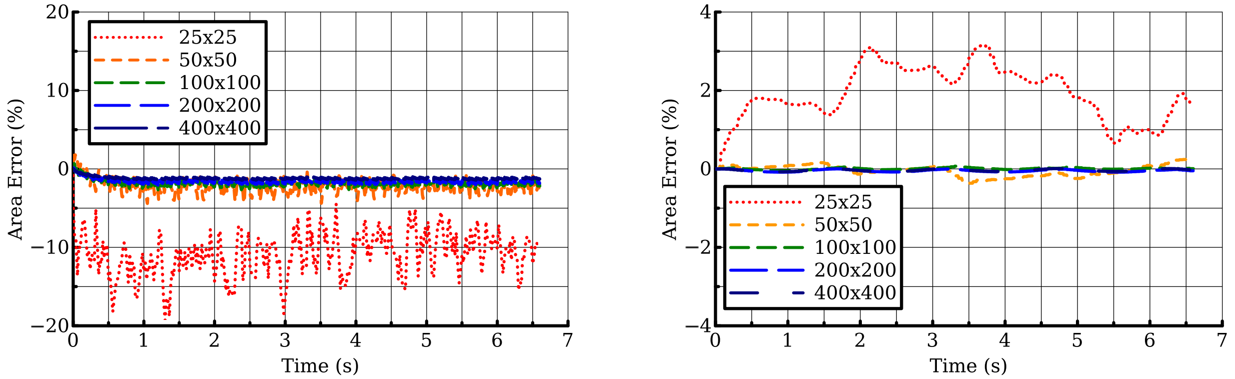
Fig. 9. Comparison of reinitialization schemes for the reinitialization of circular disc rotation problem solved on 100 × 100 grid. Each subfigure shows the 0.05, 0.5 and 0.95 level contours of the level set functions at time levels of 0, $\pi/4$, $\pi/2$ and $3\pi/4$ (left, top, right and bottom respectively).



(a) L^1 Error

(b) L^2 Error

Fig. 10. Convergence of the L^1 and L^2 error norms for the reinitialization of circular disc rotation problem solved on four (25 × 25, 50 × 50, 100 × 100 and 200 × 200) grid levels.



(a) CLS

(b) New

Fig. 11. Percentage area errors for the reinitialization of circular disc rotation problem solved on five (25 × 25, 50 × 50, 100 × 100, 200 × 200 and 400 × 400) grid levels using the CLS and the new reinitialization schemes.

Table 1
Percentage area errors, L^1 and L^2 errors computed after completion of one full rotation of different reinitialization schemes for the reinitialization of circular disc rotation problem.

Mesh	Area Error (%)		L^1 Error		L^2 Error	
	CLS	New	CLS	New	CLS	New
25×25	−5.48500	−1.95050	2.6488E−02	1.4665E−02	3.5611E−03	2.4929E−03
50×50	−2.21890	−0.15707	5.2915E−03	5.5977E−03	4.4320E−04	4.4555E−04
100×100	−1.99710	−0.02158	3.3020E−03	2.4589E−03	1.6393E−04	1.6827E−04
200×200	−1.71600	−0.00865	2.1747E−03	1.2175E−03	7.3160E−05	6.9736E−05
400×400	−1.3129	−0.00083	1.8574E−03	9.7021E−04	3.4264E−05	2.5883E−05

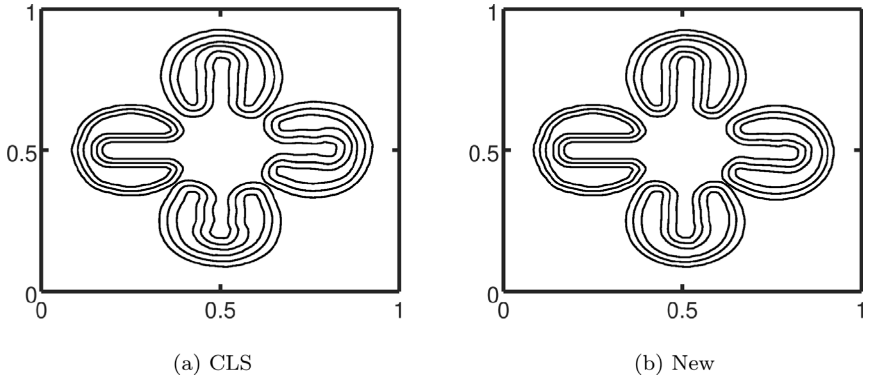


Fig. 12. Comparison of reinitialization schemes for the reinitialization of Zalesak's disc rotation problem solved on 100 × 100 grid. Each subfigure shows the 0.05, 0.5 and 0.95 level contours of the level set functions at time levels of 0.0, $\pi/4$, $\pi/2$ and $3\pi/4$ (left, top, right and bottom respectively).

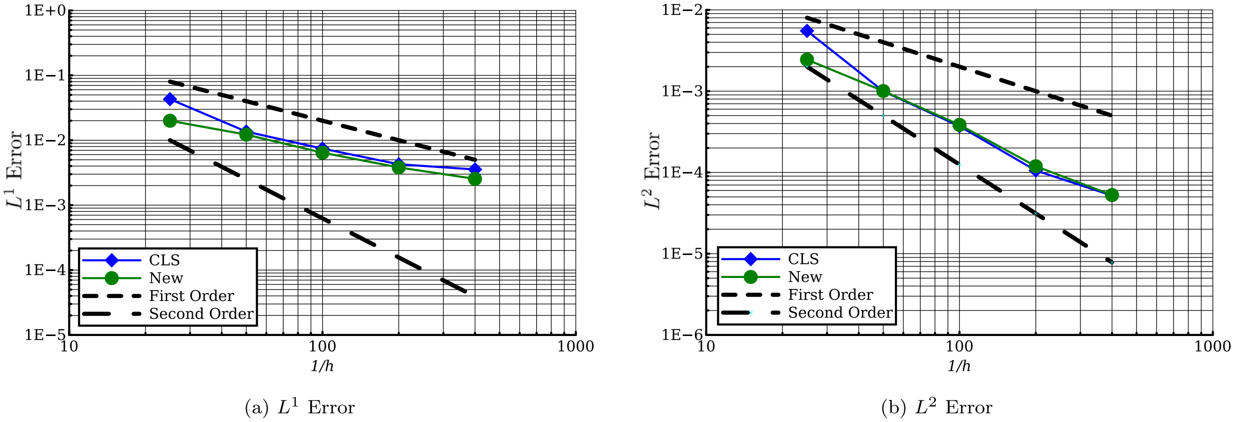


Fig. 13. Convergence of the L^1 and L^2 error norms for the reinitialization of Zalesak's disc rotation problem solved on four (25 × 25, 50 × 50, 100 × 100 and 200 × 200) grid levels.

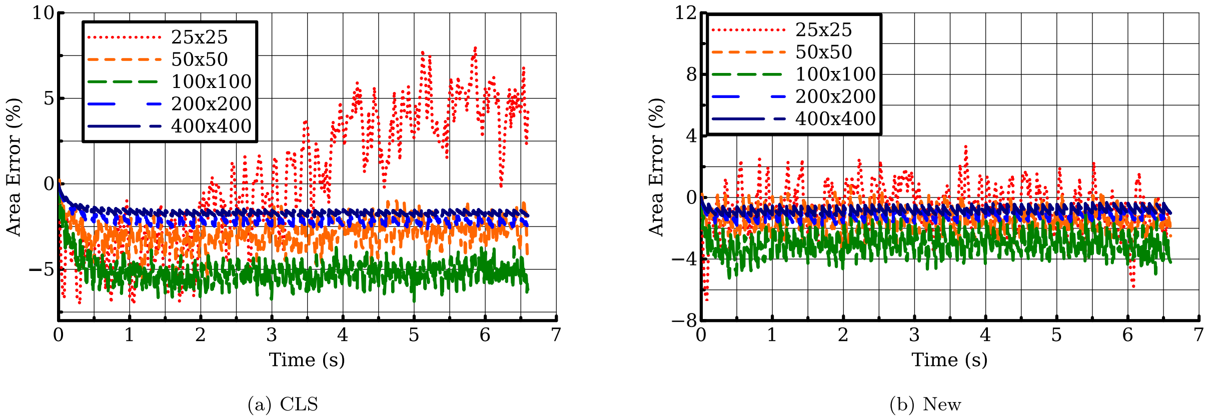
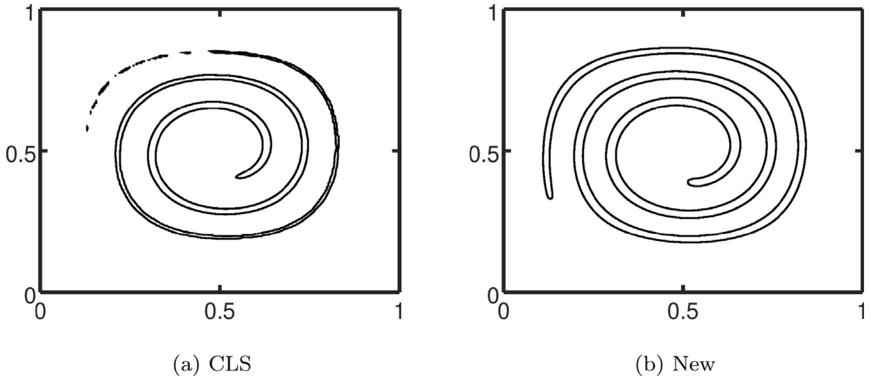


Fig. 14. Percentage area errors for the reinitialization of Zalesak's disc rotation problem solved on five (25 × 25, 50 × 50, 100 × 100, 200 × 200 and 400 × 400) grid levels using the CLS and the new reinitialization schemes.

Table 2
Percentage area errors, L^1 and L^2 errors computed after completion of one full rotation of different reinitialization schemes for the reinitialization of Zalesak's disc rotation problem.

Mesh	Area Error (%)		L^1 Error		L^2 Error	
	CLS	New	CLS	New	CLS	New
25×25	−9.49000	−7.10470	4.2972E−02	1.9959E−02	5.5200E−03	2.4447E−03
50×50	−5.33530	−3.70040	1.3590E−02	1.2202E−02	1.0000E−03	1.0036E−03
100×100	−3.09830	−1.28230	7.4338E−03	6.4325E−03	3.7050E−04	3.8000E−04
200×200	−2.38510	−0.90177	4.2656E−03	3.8105E−03	1.0556E−04	1.2000E−04
400×400	−1.81240	−0.439886	3.5455E−03	2.5418E−03	5.1928E−05	5.2658E−05



(a) CLS (b) New

Fig. 15. Comparison of reinitialization schemes for the reinitialization of circular disc deformation problem solved on 200 × 200 grid. Each subfigure shows the 0.5 level contours of the level set function at time levels of 4 s.

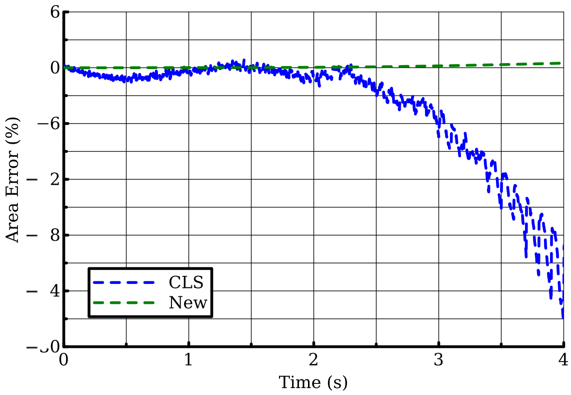


Fig. 16. Percentage area errors for the reinitialization of circular disc deformation problem solved on 200 × 200 grid levels using the CLS and the new reinitialization schemes.

reinitialization is able to capture the thin elongated tail. Fig. 16 shows the area errors plotted with respect to time for both the reinitialization approaches. Here also, one can see that the area error, in case of the new reinitialization scheme is very small compared to the CLS.

4.3. Low amplitude sloshing

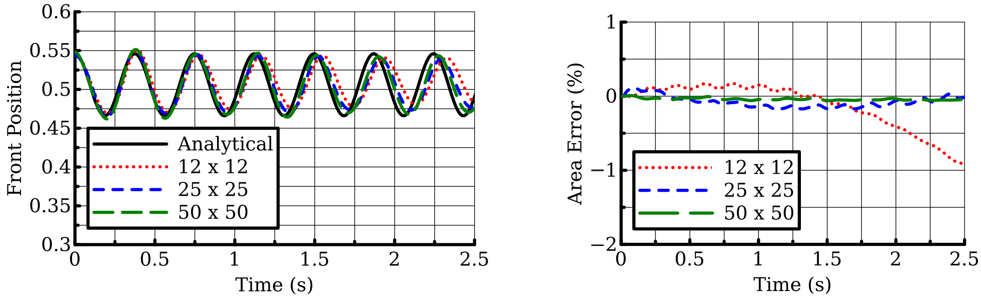
In order to quantify the performance of the new reinitialization algorithm, more realistic problems are considered next. The liquid sloshing inside a stationary water tank problem, reported in [15,18], is considered first in this category. The water tank is a square shaped one bounded between $0 \leq x \leq L$ and $0 \leq y \leq L$, where, $L = 0.1$ m. The bottom half of the tank is filled with water of density 1000 kg/m^3 and the remaining space is filled with air of density 1 kg/m^3. The air–water interface is defined initially as $y(x) = 0.05 + 0.005\cos(\pi x/L)$. The initial velocity field is set to be zero and pressure field is set based on hydrostatic conditions. Since the analytical solution is available for inviscid case, all viscous flux components for this test problem are set to zero and all boundaries are set to free-slip boundary conditions. The computational domain is discretized using uniform Cartesian meshes of coarse (12×12), medium (25×25) and fine (50×50) levels. The artificial compressibility parameter, β, is taken as 1000. The real time step, Δt, is taken as 0.01 s and after each real time step the level set function is reinitialized using a set of four reinitialization iterations. For the pseudo time iteration, the pseudo time step, $\Delta\tau$, is calculated using a Courant number, $\nu = 0.9$.

As time progresses, the water surface inside the container starts to oscillate. An approximate expression for the time period of oscillation for the first mode is obtained in [19] as, $2\pi\{g_y k \tanh(kh)\}^{-\frac{1}{2}}$, where, $h = 0.05$ m is the average depth of the water and $k = 2\pi/\lambda$ m^{-1} is the wave number with wave length $\lambda = 0.2$ m. For the present problem configuration, this time period can be estimated as 0.3739 s. In order to compare the time period of oscillation of the interface with the analytical time period, the location of the air–water interface at the left wall is measured during numerical simulations. The measured front positions are normalized with respect to the domain length (L) and plotted with respect to time, for the new reinitialization approach, in Fig. 17(a) for all the three mesh levels. One can see from Fig. 17(a) that, as the mesh is refined, the lag in the interface movement reduces and the profiles converge towards the analytical results. As described in [19], the presence of the second mode, with half time period as that of the first mode, influences the oscillation of the air–water interface by increasing the amplitude of every even peak to a higher value than that of the odd peak. This feature can also be observed from Fig. 17(a) for the fine mesh case. The area errors are quantified using Eq. (37) for all three mesh levels in the case of the test problem with the new reinitialization approach and plotted in Fig. 17(b). Here also one can see that, the area error reduces as the mesh levels get refined. The Fig. 17(a) and (b) show the mesh convergence behaviour of the new reinitialization algorithm when applied to solve incompressible two-phase flow problems. Snapshots of the air–water interface along with the velocity vectors for the fine mesh level are shown in Fig. 18.

In order to compare the results with CLS, the low amplitude sloshing problem is also solved by replacing the new reinitialization approach with the CLS approach. The air–water interface profiles and the area errors of the fine mesh level cases for the new and CLS are plotted in Fig. 19. One can see from Fig. 19(a) that the new reinitialization approach matches closely with the analytical result. Moreover, with the new reinitialization approach, the lag has been reduced. One can also see from Fig. 19(b) that the area error is less in the case of the new reinitialization approach.

4.4. Broken dam problem

Next test problem considered is a broken dam problem, reported in [20]. Here, an initial water column of height $2a$ and width a is kept at a zero velocity field and subjected to a hydrostatic pressure distribution inside a computational domain bounded between $0 \leq x \leq 4a$ and $0 \leq y \leq 4a$, where, $a = 0.25$ m. Unlike the previous test problem, here the effect of viscosity is considered. The density and dynamic viscosity for water is taken as 1000 kg/m^3 and 1×10^{-3}kg/m s respectively and



(a) Front positions with respect to time

(b) Percentage area errors with respect to time

Fig. 17. Front positions and percentage area errors with respect to time for the low amplitude sloshing problem solved on three different mesh levels using the new reinitialization approach. The analytical result in the subfigure (a) is based on [19].

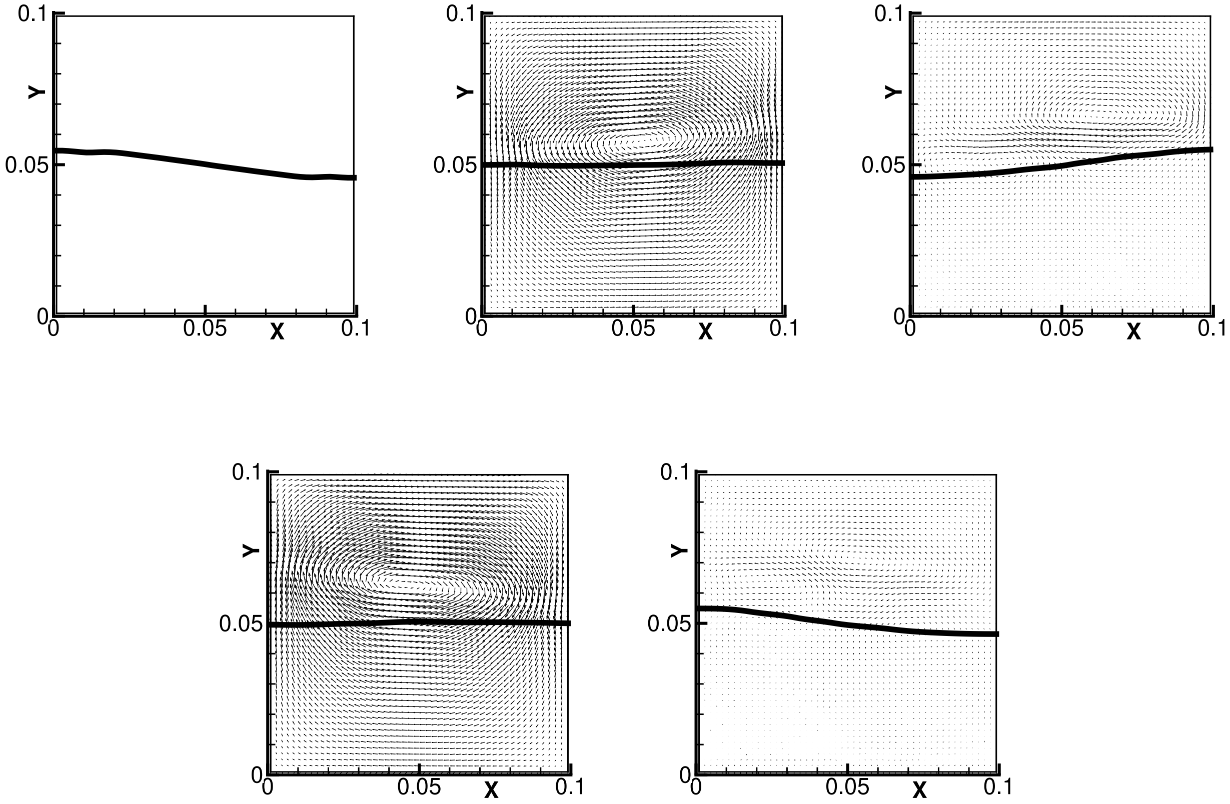
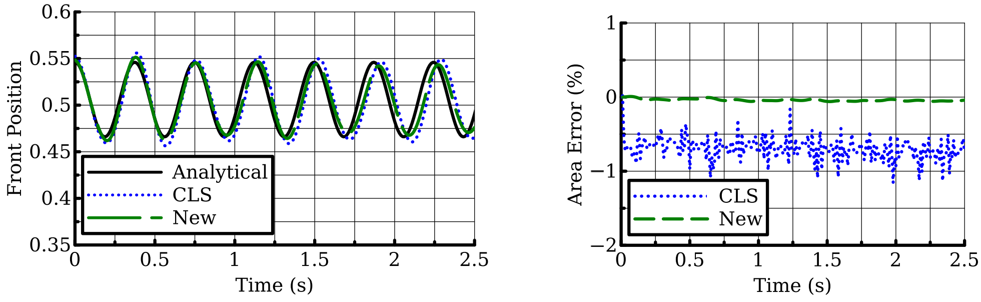


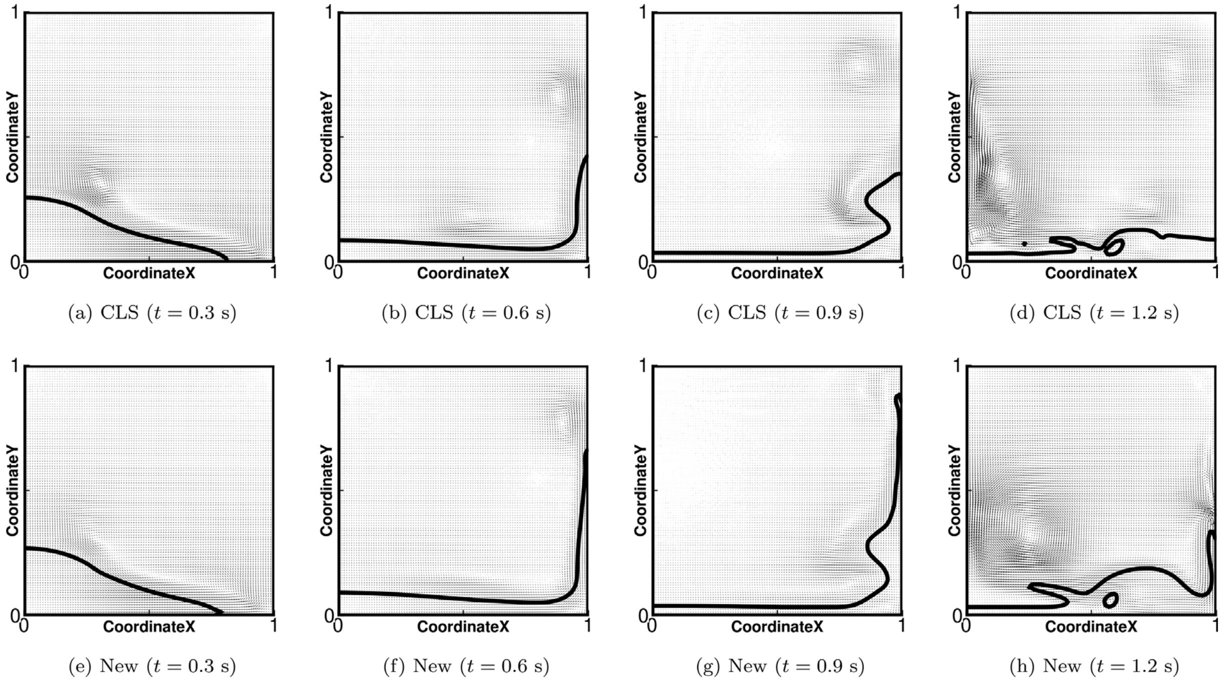
Fig. 18. Interface evolution of the low amplitude sloshing problem at different time levels from $t = 0.0$ s to $t = 0.4$ s solved on the fine level (50 × 50) mesh using the new reinitialization approach. The velocity vectors are plotted with a relative length of 0.05 grid unit per magnitude of total velocity.



(a) Front positions with respect to time

(b) Percentage area errors with respect to time

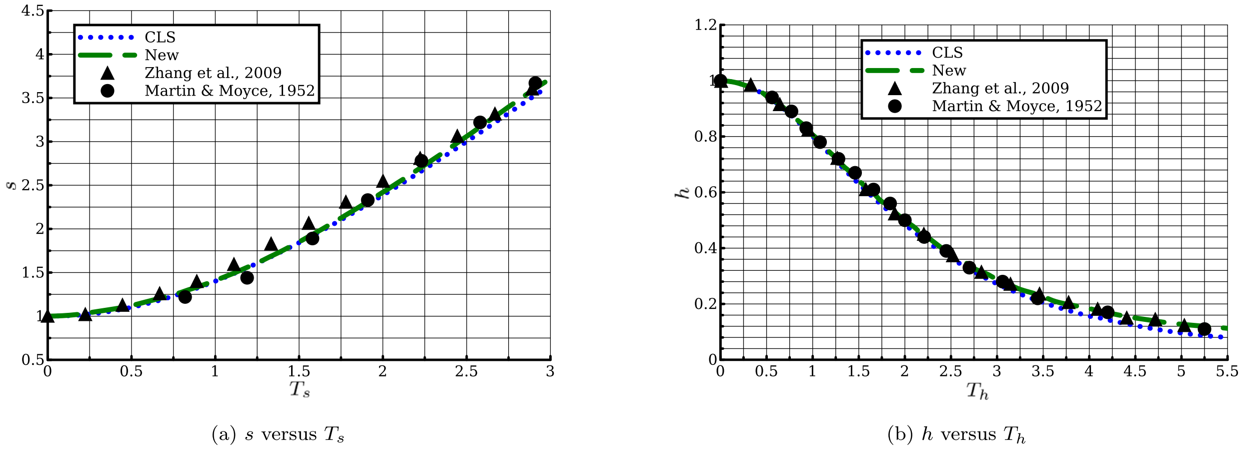
Fig. 19. Front positions and percentage area errors with respect to time for the low amplitude sloshing problem solved using the CLS and the new reinitialization approaches. The analytical result in the subfigure (a) is based on [19].



(a) CLS ($t = 0.3$ s) (b) CLS ($t = 0.6$ s) (c) CLS ($t = 0.9$ s) (d) CLS ($t = 1.2$ s)

(e) New ($t = 0.3$ s) (f) New ($t = 0.6$ s) (g) New ($t = 0.9$ s) (h) New ($t = 1.2$ s)

Fig. 20. Interface profiles at different time levels, starting from $t = 0.3$ s up to $t = 1.2$ s, for the broken dam problem. The velocity vectors are plotted with a relative length of 0.005 grid unit per magnitude of total velocity.



(a) s versus T_s (b) h versus T_h

Fig. 21. The non-dimensional surge front and water column height plotted with respect to the corresponding non-dimensional time scales for the broken dam problem.

for air as 1 kg/m^3 and 1.8×10^{-5}kg/m s respectively. All the four boundaries are set to no-slip boundary condition. The computational domain is discretized into 100 × 100 Cartesian mesh. The artificial compressibility parameter, β, is taken as 10,000. As time progresses, due to the presence of gravitational force, the water column collapses. In order to accurately capture the interface movement, a small real-time step of 0.005 s is chosen. After each real time step the level set function is reinitialized using a set of four reinitialization iterations. For stability reasons, a smaller Courant number of 0.1 is chosen for the computation of pseudo-time step.

Fig. 20 shows the snapshots comparing the air–water interface computed using the CLS and the new reinitialization procedure. One can see that the surge front, in case of the new reinitialization scheme, reaches close to the top wall before it falls to the bottom pool of water. These numerical results show close resemblance with the results reported by [20]. Whereas, in case of CLS, the thin surge front is spoiled due to inaccuracies arising from the reinitialization scheme. In order to compare the numerical results with experimental data reported in [21], the non-dimensional surge front, s, and non-dimensional water column height, h, are plotted with respect to the corresponding non-dimensional time scales $T_s = t\sqrt{2g/a}$ and $T_h = t\sqrt{g/a}$ respectively in Fig. 21. The surge front and water column heights are non-dimensionalized with respect to their initial sizes. Looking at Fig. 21, one can see an excellent match of both the numerical results with the experimental data reported by Martin and Moyce in [21]. Finally, Fig. 22 shows the percentage area errors, computed using Eq. (37), with respect to time for both the cases. From Fig. 22, one can see that the area loss is relatively high in both the cases. However, compared to the CLS, the area loss is relatively less in the case of the new reinitialization.

4.5. Rayleigh–Taylor instability problem

A Rayleigh–Taylor Instability problem similar to the one reported in [15,22] is considered next. Unlike the previous problem,

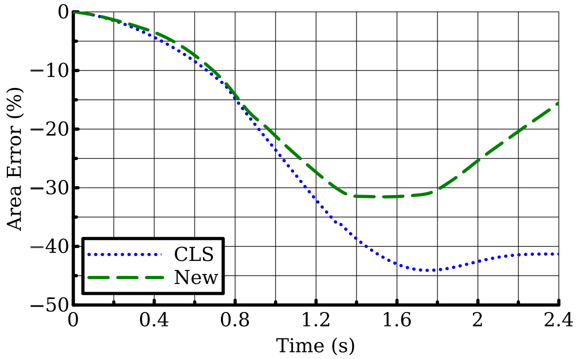


Fig. 22. The percentage area error plotted with respect to time for the broken dam problem.

viscosity plays an important role here. In this problem, a heavier fluid of density 1.225 kg/m^3 is placed on top of a lighter fluid of density 0.1694 kg/m^3 inside the computational domain bounded between $0 \leq x \leq 1$ and $0 \leq y \leq 4$. The dynamic viscosity for both the fluids are taken to be the same, $\mu_1 = \mu_2 = 3.1304952 \times 10^{-3}$ kg/m s. The two fluids are initially separated by an interface, defined as, $y = 2.0 + 0.05\cos(2\pi x)$. The problem is solved on a Cartesian mesh of 32 × 128 finite volume cells. The top and bottom boundaries are set to no-slip boundary condition, whereas, the left and right boundaries are set as symmetric boundary condition. The initial velocity field is set to be zero and the pressure field is set based on gravity. The artificial compressibility parameter, β, is taken as 1000 and the real-time step is taken as 0.01 s. After each real time step the level set function is reinitialized using a set of four reinitialization iterations. For stability reasons, the Courant number is chosen as 0.9 for the computation of the pseudo-time step.

As time progress, the top heavy fluid starts to penetrate into the bottom light fluid resulting the formation of an inverted mushroom shaped structure. Snapshots at different time levels during the evolution of the fluid–fluid interface for both the existing and new reinitialization cases are shown in Fig. 23. Similar to the previous problem, here also one can see that the new reinitialization scheme is better in capturing the thin fluid layer originating from the tips of the inverted mushroom head. Fig. 24 compares the percentage area errors computed using Eq. (37) in both the cases. One can clearly see a higher area loss for the CLS case during the later stages of interface evolution.

4.6. Rising bubble problem

The last problem considered in this section is a rising bubble problem. Unlike previous problems, this one is more challenging because of the presence of buoyancy, viscosity and surface tension forces. The rising bubble problem has been extensively investigated by Hysing et al. [23] using a set of Eulerian Finite Element based and arbitrary Lagrangian–Eulerian (ALE) type moving mesh based solvers. Apart from reporting snapshots of bubbles at different time levels, several parameters measuring the degree of deformation of the shape of the bubble and its evolution characteristics in time have been defined in [23] for the purpose of quantitative benchmarking. The reference solutions for the rising bubble problem are produced by three teams running accurate numerical simulations independently. These results therefore serve as an excellent reference for validating incompressible two-phase flow codes.

The test problem consists of a circular bubble of diameter, $d_b = 0.5$ m, placed at the lower half of a rectangular domain bounded between $0 \leq x \leq 1$ m and $0 \leq y \leq 2$ m, initially filled with a quiescent liquid. Due to the presence of the buoyant force, the initial circular bubble will start rising. With the interaction of the surrounding liquid, the initially circular shape of the bubble gets deformed. The degree of deformation of the circular bubble depends upon the Reynolds number (Re) and the Eötvös number (Eo). The Re and Eo are defined as,

$$Re = \frac{\rho_2 U_g L}{\mu_2} \tag{40}$$

and

$$Eo = \frac{\rho_2 U_g^2 L}{\sigma} \tag{41}$$

where, the characteristic length scale, $L = d_b$, and the characteristic velocity scale, $U_g = \sqrt{g d_b}$. Based on the level of difficulty, two versions of rising bubble problems are reported in [23]. The first one (denoted here as “Case-1”) is relatively simple, and the second one (denoted here as “Case-2”) is more challenging. The physical parameters defining the two dimensional rising bubble test cases are given in Table 3.

In both the cases, the initial velocity field is set to zero and the initial pressure field is set based on gravity. The left and right boundaries are set to free-slip boundary condition and the top and bottom walls are set to no-slip boundary condition. The artificial compressibility parameter, β, is taken as 10,000. The real-time step is taken as 0.05 s and after each real time step the level set function is reinitialized using a set of four reinitialization iterations. For the pseudo time iteration, the pseudo time step, $\Delta\tau$, is calculated using a Courant number, $\nu = 0.9$. Numerical simulations are carried out up to a time level of 4 s. In order to make a quantitative comparison, three parameters, namely, the rise velocity, the location of centroid and the circularity of the bubble are reported in [23]. These parameters are computed as,

$$\text{Rise Velocity, } v_c = \frac{\int_{\Omega_b} \mathbf{u} \cdot \mathbf{e}_y \, \mathrm{d}\Omega_b}{\int_{\Omega_b} \mathrm{d}\Omega_b}, \tag{42}$$

$$\text{Centroidal Location, } y_c = \frac{\int_{\Omega_b} \mathbf{x}_b \cdot \mathbf{e}_y \, \mathrm{d}\Omega_b}{\int_{\Omega_b} \mathrm{d}\Omega_b} \tag{43}$$

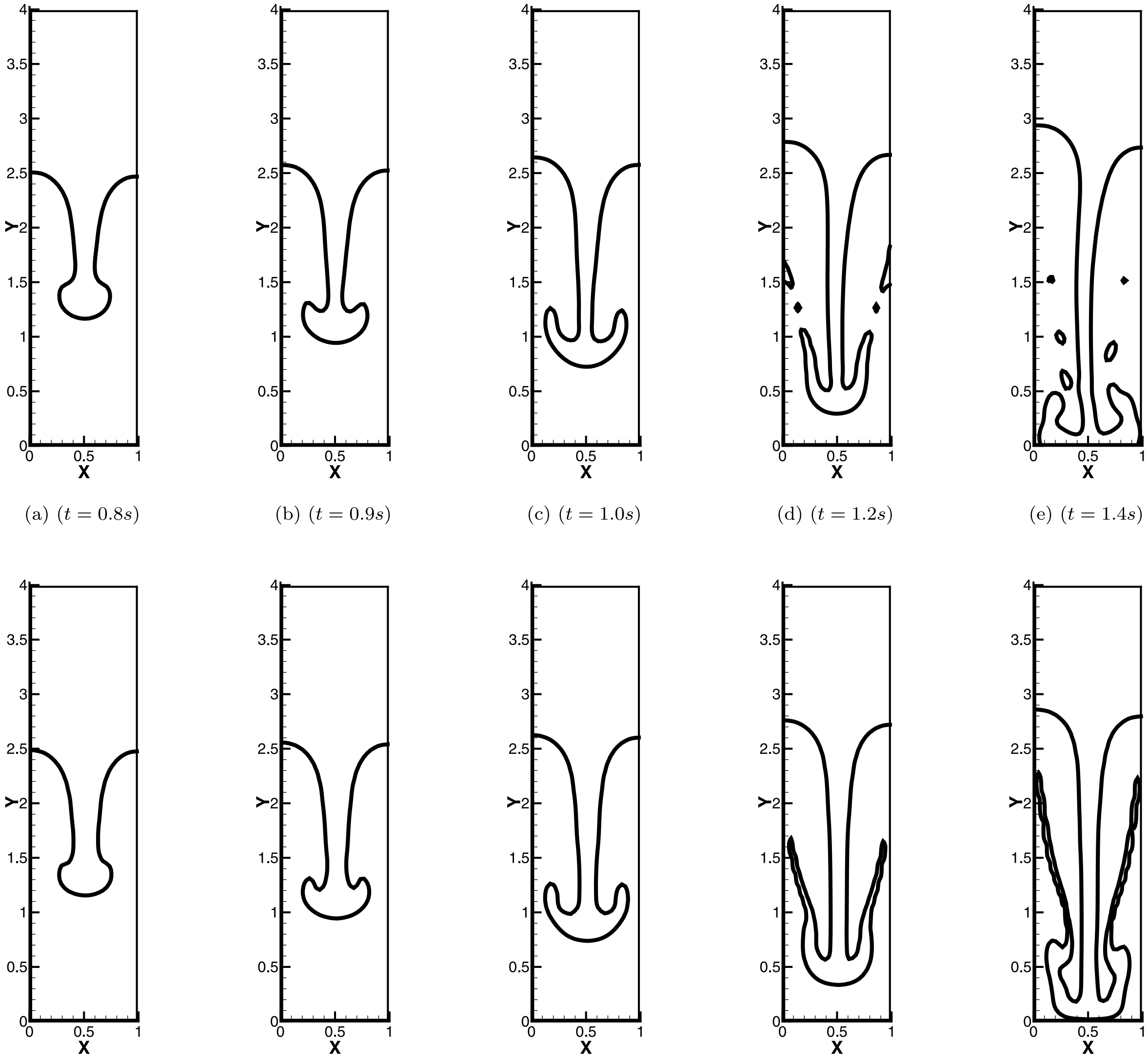
and

$$\text{Circularity, } \zeta = \frac{P_a}{P_b} = \frac{\text{Perimeter of area-equivalent bubble}}{\text{Perimeter of the bubble}} \\ = \frac{\pi d_a}{\int_{\Omega} \|\nabla\psi\| \mathrm{d}\Omega} \tag{44}$$

where, Ω is the computational domain, Ω_b is the region occupied by the bubble, $\mathbf{x}_b$ is the position vector inside the bubble, $\mathbf{e}_y$ is the unit vector parallel to the y-axis and d_a is the diameter of a circle with an area equal to that of the bubble with circumference P_b.

4.6.1. Case-1

For the choice of physical parameters of Case-1, the bubble does not undergo large deformation. The initial circular bubble first stretches in the horizontal direction and, finally, settles down to an ellipsoidal profile as it reaches its terminal speed. Numerical simulations are carried out on a Cartesian mesh of 80 × 160 finite volume cells. The bubble profiles at different time levels for both the CLS and the new reinitialization cases are shown in Fig. 25. One can see that the bubble profiles for the CLS and the new reinitialization cases are quite similar and match very well with the results reported in [23]. In order to make a close comparison, the terminal shape of bubbles in both the cases are plotted in Fig. 26 along with the reference bubble profile of [23]. One can see from Fig. 26 that the bubble profiles of both the CLS and the new



(a) $(t = 0.8s)$ (b) $(t = 0.9s)$ (c) $(t = 1.0s)$ (d) $(t = 1.2s)$ (e) $(t = 1.4s)$

(f) $(t = 0.8s)$ (g) $(t = 0.9s)$ (h) $(t = 1.0s)$ (i) $(t = 1.2s)$ (j) $(t = 1.4s)$

Fig. 23. Interface profiles at different time levels, starting from $t = 0.8$ s up to $t = 1.4$ s, for the Rayleigh–Taylor instability problem. Subfigures from (a) to (e) correspond to the CLS scheme and from (f) to (j) correspond to the new reinitialization scheme.

Table 3
Physical parameters defining the two dimensional rising bubble test cases.

Test case	ρ_2	ρ_1	μ_2	μ_1	g	σ	Re	Eo	ρ_2/ρ_1	μ_2/μ_1
Case-1	1000	100	10	1	0.98	24.5	35	10	10	10
Case-2	1000	1	10	0.1	0.98	1.96	35	125	1000	100

reinitialization schemes match very well with the reference bubble profile. The rise velocity, centroid location and the circularity of the rising bubble are plotted with respect to time in Fig. 27. Here also, both the CLS and the new reinitialization results match closely with the reference plots. Finally, the percentage area error, computed using Eq. (37), is plotted with respect to time in Fig. 28. It can be noticed that the area error is relatively less for the new reinitialization case compared to that of CLS.

4.6.2. Case-2

Unlike the previous case, a large density ratio in this case results the bubble to deform more and acquire a dimple cap profile with thin elongated filament like structures originating from both sides. Due to the complex shape, it is relatively difficult to capture the bubble profile in Case-2 as compared to the Case-1. Numerical simulations are carried out on a Cartesian mesh of 80 × 160 finite volume cells. The snapshots of bubble profiles at different time levels for both the CLS and the new reinitialization cases are shown in Fig. 29. One can clearly see from Fig. 29 that the elongated filament structure is not captured very well in case of the CLS case. Whereas, a better profile of the elongated filament structure is captured in case of the new reinitialization method. The rise velocity, centroid location and circularity of the bubble are plotted with respect to time in Fig. 30. One can see that the results for the new reinitialization scheme show better match with the reference results. Finally, percentage area errors, computed using Eq. (37), are plotted in Fig. 31. Here also, one can see that, the area error for the new reinitialization scheme is less compared to the area error in case of the CLS.

Finally, in order to demonstrate the ability of the new reinitialization scheme to deal with complex meshes, the problem is also solved on an unstructured mesh consisting of 23,331 finite

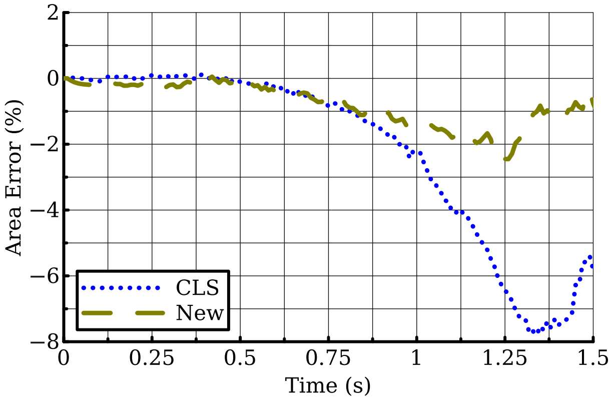


Fig. 24. The percentage area error plotted with respect to time for the Rayleigh–Taylor instability problem.

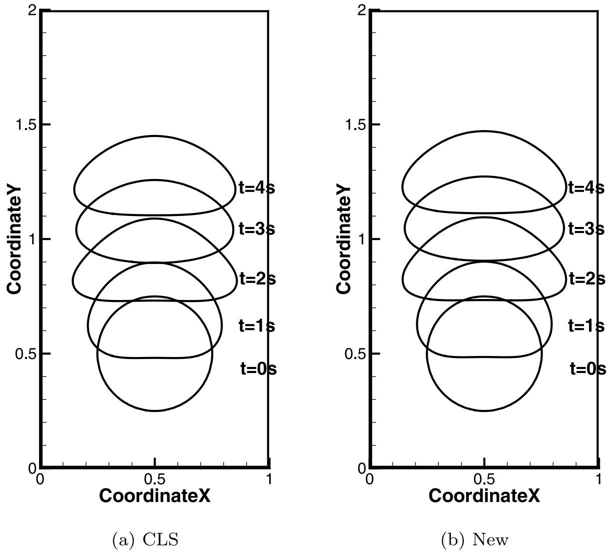


Fig. 25. Bubble profiles from $t = 0$ s up to $t = 4$ s for the rising bubble problem (Case-1).

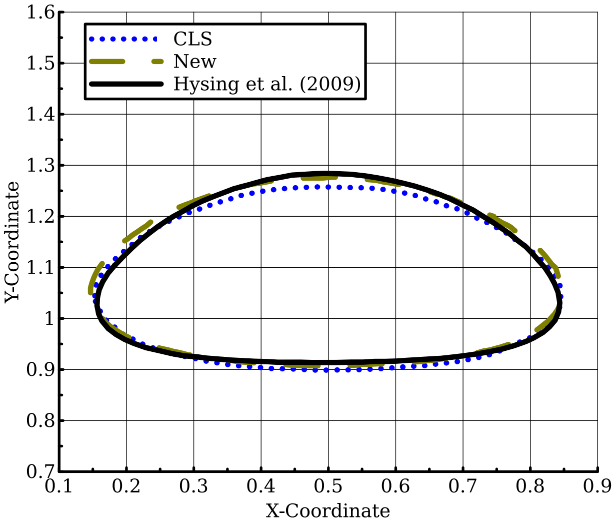


Fig. 26. Enlarged bubble profiles at $t = 3$ s for the rising bubble problem (Case-1).

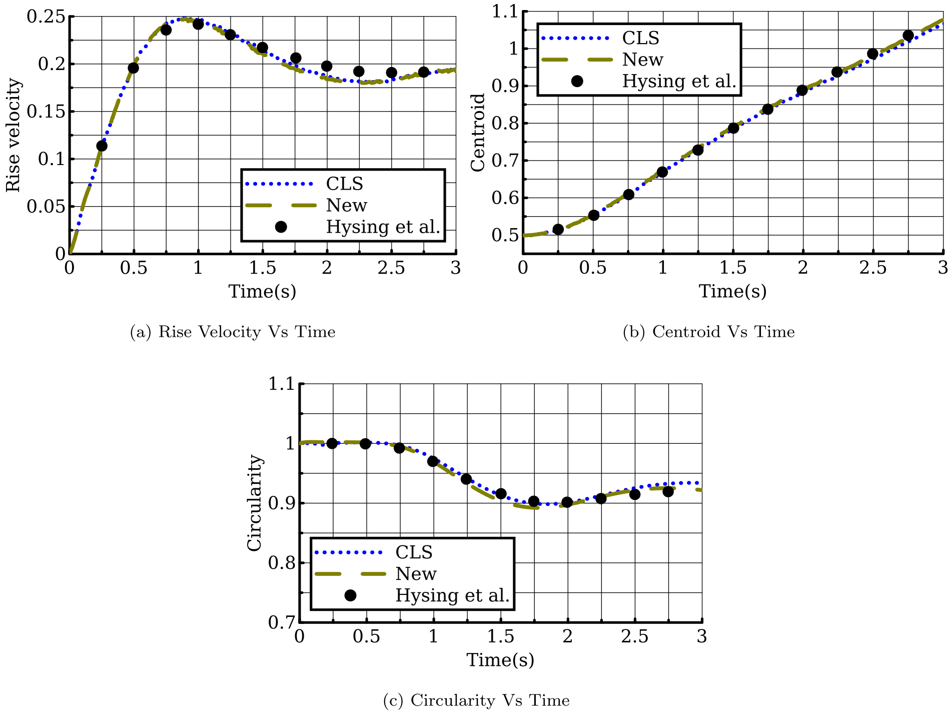
volume cells of triangular and quadrilateral shapes. Fig. 32 shows the unstructured meshes considered for this case. Due to the clustering of cells in the bubble path, one may expect a better result in this case compared to the structured mesh case. The snapshots of bubble profiles at different time levels for the new reinitialization case on unstructured mesh are shown in Fig. 33. One can see from Fig. 33 that the elongated filament structure is better captured in this case. The bubble profiles captured using the new reinitialization approach solved on unstructured mesh show a close resemblance with the fine mesh results reported in [23]. The rise velocity, centroid location and circularity of the bubble are plotted with respect to time in Fig. 34. One can see that the results for the new reinitialization scheme on the unstructured mesh show very good match with the reference results. Finally, the percentage area errors, computed using Eq. (37), are plotted in Fig. 35. In comparison with the structured mesh results, the new reinitialization method solved on unstructured mesh shows much less area error.

5. Conclusion

A new approach to reinitialize the level set function for the CLS method is formulated in this paper. To start with, the existing artificial compression based reinitialization equation is examined carefully in order to identify the term responsible for the unwanted movement of the interface contour. After isolating and removing a curvature dependent velocity term, that is responsible for wrongly moving the interface contour, the reinitialization equation is revised. The remaining terms in the reinitialization equation are then carefully replaced with equivalent terms that do not involve contour normal vectors. Unlike the compression and diffusion fluxes present in a typical artificial compression approach, the newly reformulated approach has a level set sharpening term, responsible for the narrowing the level set profile, and a balancing term in order to counteract the effect of sharpening. The combined effect of sharpening and balancing restores the level set function without causing any unwanted movement to the interface contour. Due to the absence of terms involving contour normal vectors, the susceptibility towards the formation of unphysical fluid patches during the reinitialization process has been completely eliminated in the new reinitialization procedure.

In order to evaluate the performance of the new reinitialization approach, three types of numerical test cases are carried out. The in-place reinitialization problems demonstrate the interface-preserving property and the superior convergence behaviour of the new reinitialization approach. The area and shape errors of the new approach are quantified using a set of scalar advection based test problems. To evaluate the performance on more practical problems, a set of standard incompressible two-phase flow problems, starting from an inviscid test case to complex test cases involving viscous and surface tension forces, are solved. Finally, in order to demonstrate the ability to deal with complex mesh types, an incompressible two-phase flow problem is also solved on an unstructured mesh consisting of finite volume cells having triangular and quadrilateral shapes. Following are some of the attractive features of the new reinitialization approach:

- The new reinitialization approach does not suffer from the two major drawbacks of the existing artificial compression based reinitialization procedure, namely, the unwanted curvature-dependent movement of the interface contour and strong sensitivity towards numerical errors leading to the formation of unphysical fluid patches. As a result, even a very thin and elongated fluid element like structures in two-phase flow problem can be easily captured.



(a) Rise Velocity Vs Time

(b) Centroid Vs Time

(c) Circularity Vs Time

Fig. 27. The rise velocity, centroid location and circularity plotted with respect to time for the rising bubble problem (Case-1).

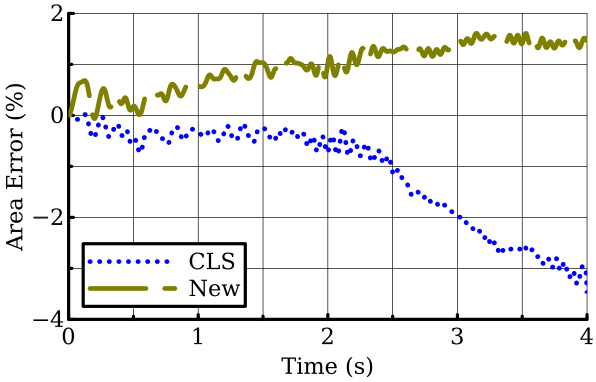


Fig. 28. The percentage area error plotted with respect to time for the rising bubble problem (Case-1).

- The steady state form of the new reinitialization equation exactly satisfies the hyperbolic tangent profile of the level set function. The method is stable and has better numerical convergence properties.
- With less complex terms in the new reinitialization equation, one can choose a larger time step, approximately by a factor of $4/h$, in comparison with the allowable time step of an artificial compression based approach. In addition, each reinitialization iteration of the new reinitialization approach is around 17 times faster than that of the original CLS approach, aiding an overall reduction in computational efforts.
- The new framework is applicable to unstructured meshes consisting of finite volume cells having triangular and quadrilateral shapes, increasing the scope for solving a wide range of practical problems.

With enhanced accuracy and improved ability to deal with complex mesh types, the proposed reinitialization approach can be efficiently used in solving real life incompressible two-phase flow problems.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgements

The present work is partially supported by Aeronautics Research & Development Board (AR&DB), India, with the project Grant number ARDB/01/1031930/M/I DT. 13.09.2019. We gratefully thank AR&DB for the support.

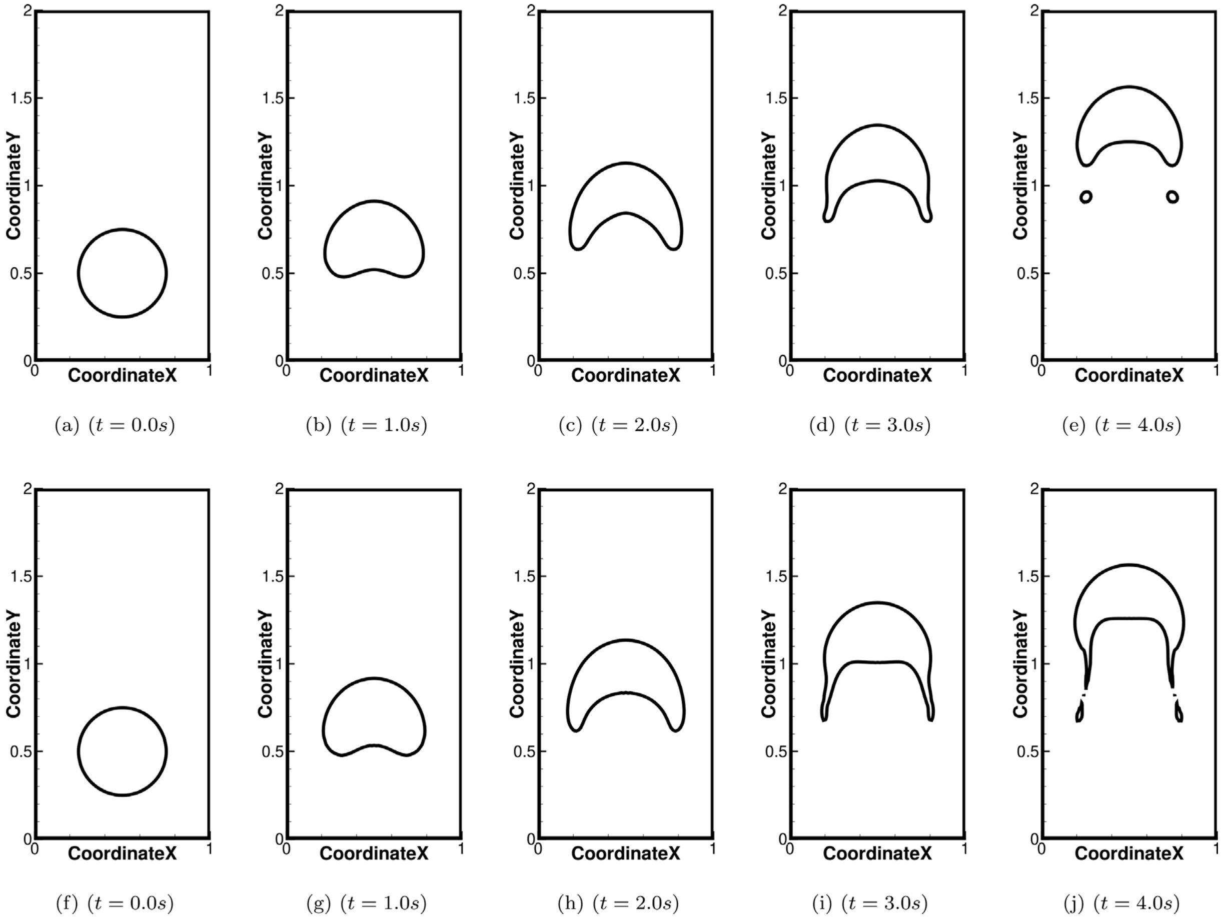
Appendix A. Central least square estimation of level set gradients

The second term in Eq. (17) involves computation of level set gradient terms.

$$|\nabla\psi|_i = \sqrt{\left(\frac{\partial\psi}{\partial x}\right)_i + \left(\frac{\partial\psi}{\partial y}\right)_i} \tag{A.1}$$

These terms are evaluated here using a central least square approach. In order to construct cell centre derivatives, a stencil consisting of vertex based neighbours, as shown in Fig. A.36, is considered. Using Taylor series expansion, the neighbour cell values, ψ_j, of the level set function are expressed in terms of the value at the cell i, as,

$$\psi_j = \psi_i + (x_j - x_i)\left(\frac{\partial\psi}{\partial x}\right)_i + (y_j - y_i)\left(\frac{\partial\psi}{\partial y}\right)_i \dots \tag{A.2}$$



(a) $(t = 0.0s)$ (b) $(t = 1.0s)$ (c) $(t = 2.0s)$ (d) $(t = 3.0s)$ (e) $(t = 4.0s)$

(f) $(t = 0.0s)$ (g) $(t = 1.0s)$ (h) $(t = 2.0s)$ (i) $(t = 3.0s)$ (j) $(t = 4.0s)$

Fig. 29. The bubble profiles at different time levels starting from $t = 0$ s up to $t = 4$ s for the rising bubble problem (Case-2) solved on 80 × 160 Cartesian mesh. The subfigures from (a) to (e) correspond to the CLS case and from (f) to (j) correspond to the new reinitialization case.

where, (x_i, y_i) and (x_j, y_j) are locations of the centroids of cell i and centroid of the neighbour cell j respectively. Upon truncating higher order terms (after the third order term) and re-arranging, Eqs. (A.2) can be written as,

$$\Delta\psi = \mathbf{S}\,\mathrm{d}\psi \tag{A.3}$$

where,

$$\Delta\psi = \begin{Bmatrix} \psi_1 - \psi_i \\ \psi_2 - \psi_i \\ \dots \\ \dots \\ \psi_l - \psi_i \end{Bmatrix}; \quad \mathbf{S} = \begin{bmatrix} x_1 - x_i & y_1 - y_i \\ x_2 - x_i & y_2 - y_i \\ \dots & \dots \\ \dots & \dots \\ x_l - x_i & y_l - y_i \end{bmatrix};$$

$$\mathrm{d}\psi = \begin{Bmatrix} \frac{\partial\psi}{\partial x} \\ \\ \frac{\partial\psi}{\partial y} \end{Bmatrix}$$

; ; The overdetermined system of Eq. (A.3) can be solved as,

$$\mathrm{d}\psi = \left(\mathbf{S}^{\mathrm{T}}\mathbf{S}\right)^{-1}\mathbf{S}^{\mathrm{T}}\Delta\psi \tag{A.4}$$

Closed-form expressions for the derivatives can be obtained by simplifying equation (A.4) as,

$$\left(\frac{\partial\psi}{\partial x}\right)_i = \frac{\ell_{22}r_1 - \ell_{21}r_2}{G} \tag{A.5a}$$

$$\left(\frac{\partial\psi}{\partial y}\right)_i = \frac{\ell_{11}r_2 - \ell_{12}r_1}{G} \tag{A.5b}$$

where

$$\ell_{11} = \sum_{j=1}^{l}(x_j - x_i)^2, \qquad \ell_{22} = \sum_{j=1}^{l}(y_j - y_i)^2,$$

$$\ell_{12} = \ell_{21} = \sum_{j=1}^{l}(x_j - x_i)(y_j - y_i)$$

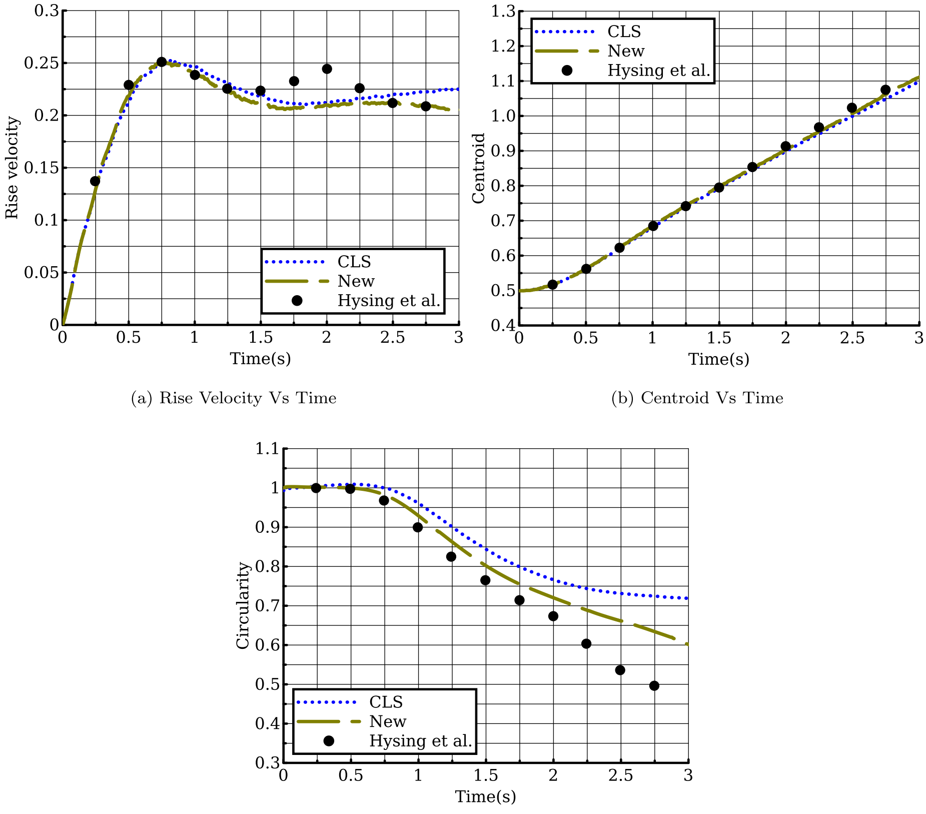
$$r_1 = \sum_{j=1}^{l}(x_j - x_i)(\psi_j - \psi_i), \qquad r_2 = \sum_{j=1}^{l}(y_j - y_i)(\psi_j - \psi_i),$$

$$G = \ell_{11}\ell_{22} - \ell_{12}^2$$

Appendix B. Overview of existing improvements in the reinitialization scheme

B.1. Efforts towards arresting undesired movement of interface contour

In order to alleviate the problem of undesired movement of interface contours, localization of the reinitialization process is suggested in the literature. In [7], a localized version of the artificial compression procedure (the Localized CLS (LCLS) approach) is developed and used along with a Constrained Interpolation Profile (CIP) method. Here, a parameter, β, is introduced in order to vary the degree of interface compression and dissipation over



(a) Rise Velocity Vs Time

(b) Centroid Vs Time

(c) Circularity Vs Time

Fig. 30. The rise velocity, centroid location and circularity plotted with respect to time for the rising bubble problem (Case-2).

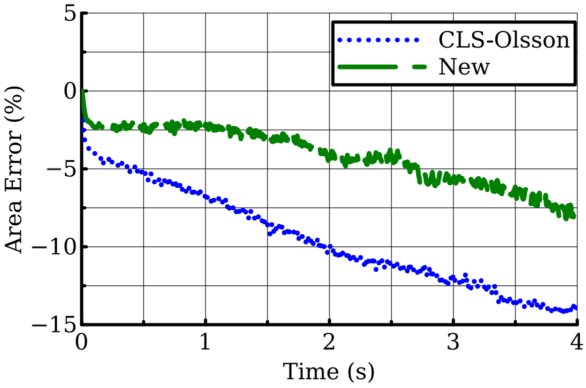


Fig. 31. The percentage area error plotted with respect to time for the rising bubble problem (Case-2).

the domain. The local sharpening equation, as reported in [7], can be written as,

$$\frac{\partial \psi}{\partial \tau_r} + \beta \nabla \cdot (\psi(1-\psi)\mathbf{n}) = \beta \varepsilon \nabla \cdot (\nabla \psi) \tag{B.1}$$

One can see that the key difference between Eq. (B.1) and the original reinitialization equation by Olsson et al. in [1] is the introduction of a new term β multiplied to the compression and diffusion terms. The term β depends upon a diffusive index, α, of ψ, as,

$$\beta = \begin{cases} \beta_{\min} & \text{for } \alpha \le 0, \\ \frac{1-\beta_{\min}}{\alpha_{\text{sat}}}\alpha + \beta_{\min} & \text{for } 0 < \alpha \le \alpha_{\text{sat}}, \\ 1 & \text{for } \alpha \ge \alpha_{\text{sat}}, \end{cases} \tag{B.2}$$

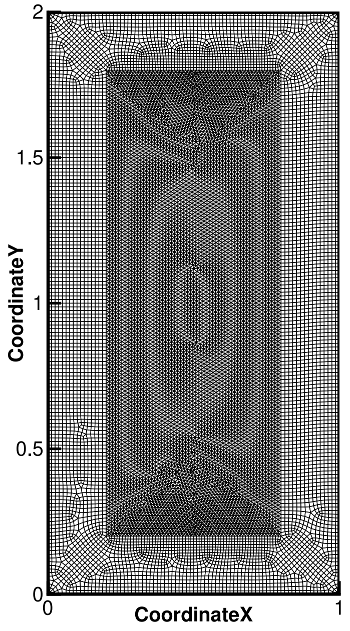
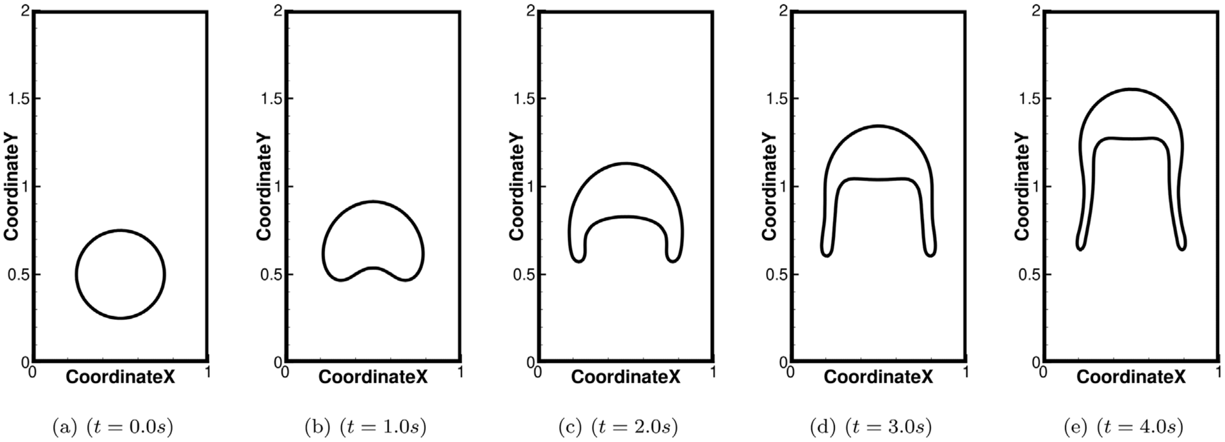


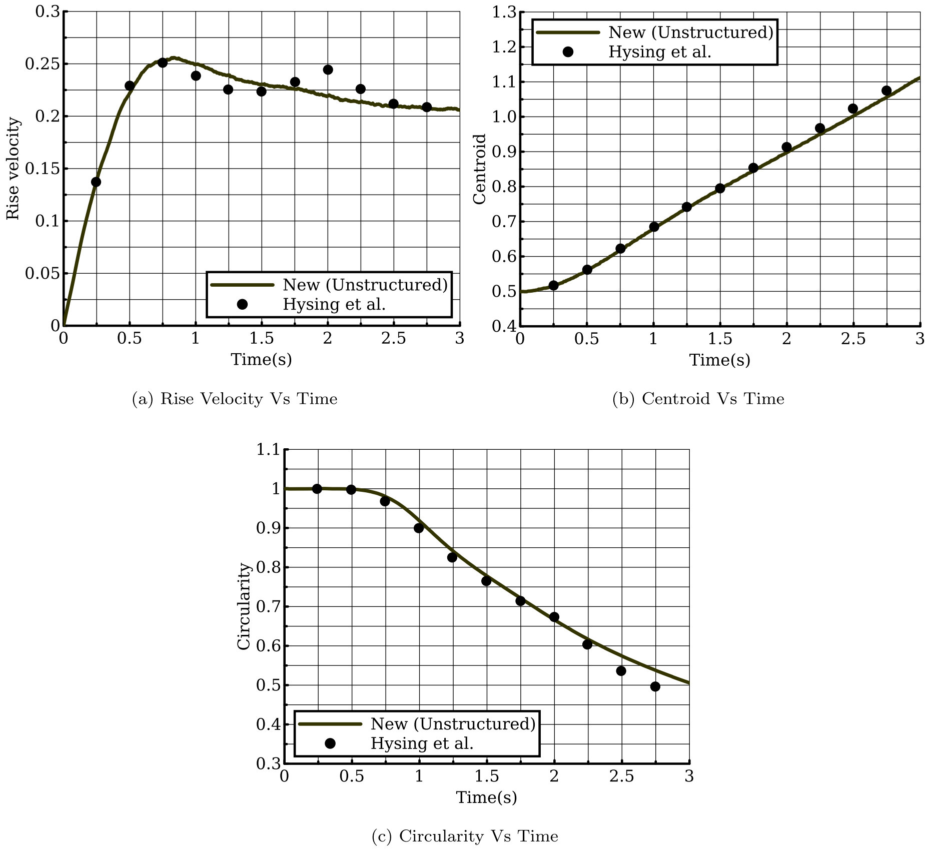
Fig. 32. The unstructured mesh used for solving the rising bubble problem (Case-2). The mesh consists of a total of 23,331 finite volume cells with triangular and quadrilateral shapes.

where, the diffusive index, α is computed as $\alpha = (\nabla(\vec{u} \cdot \mathbf{n}) \cdot \mathbf{n})$. The parameters α_{sat} and $\beta_{\min}$ are obtained based on numerical experiments. Through the parameter, β, the degree of reinitialization (that is, the amount of compression and diffusion) in this



(a) $(t = 0.0s)$ (b) $(t = 1.0s)$ (c) $(t = 2.0s)$ (d) $(t = 3.0s)$ (e) $(t = 4.0s)$

Fig. 33. The bubble profiles at different time levels starting from $t = 0$ s up to $t = 4$ s for the rising bubble problem (Case-2) solved on unstructured mesh.



(a) Rise Velocity Vs Time

(b) Centroid Vs Time

(c) Circularity Vs Time

Fig. 34. The rise velocity, centroid location and circularity plotted with respect to time for the rising bubble problem (Case-2).

approach is varied over the computational domain. In regions where $\alpha \leq 0$, the degree of reinitialization will be significantly low.

The localization procedure developed in [4] is, however, different from [7]. Unlike the β parameter in [7], here the compression and diffusion terms are multiplied using a spatially and temporarily varying factor $\tilde{\alpha}$. Upon expanding, this $\tilde{\alpha}$ will get absorbed into the pseudo-time variable, τ_r, and the reinitialization equation can be written as,

$$\frac{\partial \psi}{\partial \tilde{\tau}_r} + \nabla \cdot \left[(\psi(1-\psi)\mathbf{n}) - (\varepsilon\,(\nabla\psi\cdot\mathbf{n})\;\mathbf{n}) \right] = 0 \tag{B.3}$$

where, $\tilde{\tau}_r = \tilde{\alpha}\tau_r$, and the $\tilde{\alpha}$ is computed by solving $\mathbf{n}\cdot\nabla\tilde{\alpha} = 0$ using a fast marching method.

B.2. Efforts towards removing unphysical fluid patch formation far away from the interface

One of the initial attempt to overcome the issue of the unphysical fluid patch formation far away from the interface is to compute a better conditioned contour normal vector. Instead of computing the contour normal vector from the ψ-field as suggested in the artificial compression based reinitialization approach of Olsson et al. [2], the Accurate Conservative Level Set (ACLS) method [5] computes the contour normal vectors from the standard signed distance function (ϕ). Here, the ϕ-field is computed from the ψ-field using a standard fast marching method.

In the reinitialization procedure of [3], the gradient of the compression and diffusion terms are directly projected onto the

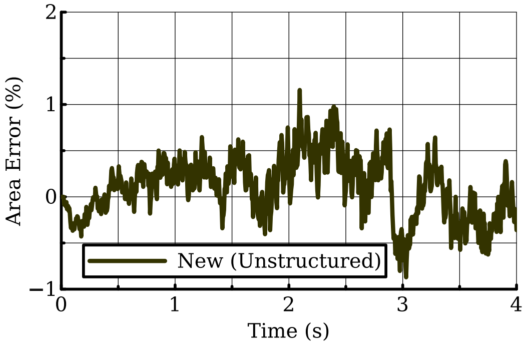


Fig. 35. The percentage area error plotted with respect to time for the rising bubble problem (Case-2).

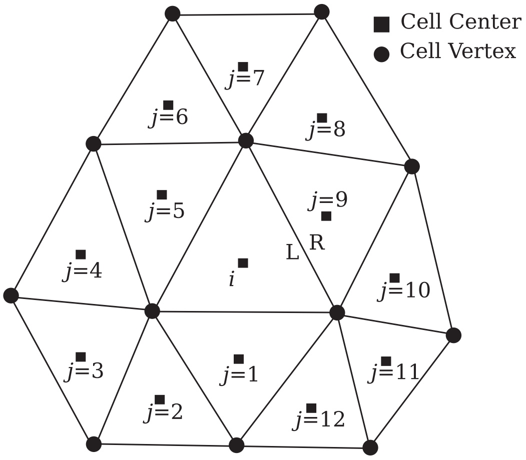


Fig. A.36. A schematic representation of a triangular shaped finite volume cell i and its vertex based neighbours, denoted as j.

contour normal vector directions, as,

$$\frac{\partial \psi}{\partial \tau_r} = \mathbf{n} \cdot \nabla \Big[\varepsilon |\nabla \psi| - \psi(1-\psi) \Big] \tag{B.4}$$

Further, the contour normal vectors are computed using a mapping function $\phi(\alpha)$, as $\mathbf{n} = \frac{\nabla \phi(\alpha)}{|\nabla \phi(\alpha)|}$. The mapping function $\phi(\alpha)$ is defined as,

$$\phi(\alpha) = \frac{\psi^\alpha}{\psi^\alpha + (1-\psi)^\alpha} \tag{B.5}$$

where, the parameter $\alpha = 0.1$.

In [10], a reformulated version of the artificial compression based reinitialization procedure of Olsson et al. [2] is developed. Upon differentiating Eq. (1) in space and taking dot product with $\mathbf{n}$ (where, $\mathbf{n} = \frac{\nabla \psi}{|\nabla \psi|} = \frac{\nabla \phi}{|\nabla \phi|}$), we can write,

$$\nabla \psi \cdot \mathbf{n} = \frac{\psi(1-\psi)}{\varepsilon} |\nabla \phi| \tag{B.6}$$

In [10], Eq. (B.6) is substituted in the original reinitialization equation by Olsson et al. in [2], to get the reformulated equation,

$$\frac{\partial \psi}{\partial \tau_r} = \nabla \cdot [\psi(1-\psi)(|\nabla \phi| - 1)\mathbf{n}] \tag{B.7}$$

Further, in order to have a better convergence property, the ϕ in Eq. (B.7) is replaced with a mapping function, ϕ_{map}, defined as,

$$\phi_{\mathrm{map}} = \frac{(\psi + \epsilon)^\gamma}{(\psi + \epsilon)^\gamma + (1 - \psi + \epsilon)^\gamma} \tag{B.8}$$

where, the parameters ϵ and γ are obtained through a set of numerical experiment in [10] as, $5 \cdot 10^{-15}$ and 10^{-5} respectively. The final reinitialization equation of [10] can be written as,

$$\frac{\partial \psi}{\partial \tau_r} = \nabla \cdot [\psi(1-\psi)(|\nabla \phi_{\mathrm{map}}| - 1)\mathbf{n}_\Gamma] \tag{B.9}$$

where, $\mathbf{n}_\Gamma = \nabla \phi_{\mathrm{map}} / |\nabla \phi_{\mathrm{map}}|$.

In [11], two modifications of [10] are proposed. Firstly, a static signed distance function computed from a fast marching method is used for the computation of the contour normal vector instead of the ϕ_{map} as used in [10]. Secondly, in order to make sure that the ψ-field is always bounded between 0 and 1 while using with non-TVD schemes, the approach in [11] replaces the $\psi(1-\psi)$ term given in [10] with $\frac{1}{4\cosh^2(\frac{\phi_{\mathrm{inv}}}{2\varepsilon})}$, where, ϕ_{inv} is computed by taking the inverse of Eq. (1) as, $\phi_{\mathrm{inv}} = \varepsilon \ln\Big(\frac{\psi}{1-\psi}\Big)$.

Table B.4
Summary of the existing and the new reinitialization formulations.

Method	Reinitialization equation	Additional equations
CLS [2] (2007)	$\frac{\partial \psi}{\partial \tau_r} = \nabla \cdot \big[(\varepsilon\, (\nabla \psi \cdot \mathbf{n})\;\; \mathbf{n}) - (\psi(1-\psi)\mathbf{n})\big]$	$\mathbf{n} = \frac{\nabla \psi_0}{\|\nabla \psi_0\|}$
ACLS [5] (2008)	$\frac{\partial \psi}{\partial \tau_r} = \nabla \cdot \big[(\varepsilon\, (\nabla \psi \cdot \mathbf{n})\;\; \mathbf{n}) - (\psi(1-\psi)\mathbf{n})\big]$	$\mathbf{n} = \frac{\nabla \phi_{\mathrm{FMM}}}{\|\nabla \phi_{\mathrm{FMM}}\|}$
CLS [3] (2010)	$\frac{\partial \psi}{\partial \tau_r} = \mathbf{n} \cdot \nabla \Big[\varepsilon \|\nabla \psi\| - \psi(1-\psi)\Big]$	$\mathbf{n} = \frac{\nabla \phi(\alpha)}{\|\nabla \phi(\alpha)\|}$; $\phi(\alpha) = \frac{\psi^\alpha}{\psi^\alpha + (1-\psi)^\alpha}$
LCLS [7] (2012)	$\frac{\partial \psi}{\partial \tau_r} = \beta \varepsilon \nabla \cdot (\nabla \psi) - \beta \nabla \cdot (\psi(1-\psi)\mathbf{n})$	$\mathbf{n} = \frac{\nabla \psi}{\|\nabla \psi\|}$; $\beta = f(\alpha)$
LCLS [4] (2014)	$\frac{\partial \psi}{\partial \tilde{\tau}_r} = \nabla \cdot \big[(\varepsilon\, (\nabla \psi \cdot \mathbf{n})\;\; \mathbf{n}) - (\psi(1-\psi)\mathbf{n})\big]$	$\mathbf{n} = \frac{\nabla \psi}{\|\nabla \psi\|}$; $\tilde{\tau}_r = \tilde{\alpha} \tau_r$; $\mathbf{n} \cdot \nabla \tilde{\alpha} = 0$
CLS [10] (2015)	$\frac{\partial \psi}{\partial \tau_r} = \nabla \cdot \big[\psi(1-\psi)(\|\nabla \phi_{\mathrm{map}}\| - 1)\mathbf{n}_\Gamma\big]$	$\mathbf{n}_\Gamma = \frac{\nabla \phi_{\mathrm{map}}}{\|\nabla \phi_{\mathrm{map}}\|}$; $\phi_{\mathrm{map}} = \frac{(\psi+\epsilon)^\gamma}{(\psi+\epsilon)^\gamma + (1-\psi+\epsilon)^\gamma}$
CLS [11] (2017)	$\frac{\partial \psi}{\partial \tau_r} = \nabla \cdot \Bigg[\frac{1}{4\cosh^2\left(\frac{\phi_{\mathrm{inv}}}{2\varepsilon}\right)}(\|\nabla \phi_{\mathrm{inv}} \cdot \mathbf{n}\| - 1)\mathbf{n}\Bigg]$	$\mathbf{n} = \frac{\nabla \phi_{\mathrm{FMM}}}{\|\nabla \phi_{\mathrm{FMM}}\|}$; $\phi_{\mathrm{inv}} = \varepsilon \ln\left(\frac{\psi}{1-\psi}\right)$
SCLS [8] (2018)	$\frac{\partial \psi}{\partial \tau_r} = -\nabla \cdot \Big((\psi(1-\psi)\mathbf{m}) + (\varepsilon(\nabla \psi \cdot \mathbf{m})\mathbf{m}) + ((1-\|\mathbf{m}\|^2)\varepsilon \nabla \psi)\Big)$	$\mathbf{m} = \frac{\varepsilon \nabla \psi}{\Big(\varepsilon^2 \|\nabla \psi\|^2 + \alpha^2 \exp(-\beta \varepsilon^2 \|\nabla \psi\|^2)\Big)^{1/2}}$
New	$\frac{\partial \psi}{\partial \tau_n} = \theta\,[-\psi(1-\psi)(1-2\psi) + \varepsilon(1-2\psi)\|\nabla \psi\|]$	No additional equations

The reinitialization equation as proposed in [11] can be written as,

$$\frac{\partial \psi}{\partial \tau_r} = \nabla \cdot \left[\frac{1}{4\cosh^2\left(\frac{\phi_{\text{inv}}}{2\varepsilon}\right)} (|\nabla \phi_{\text{inv}} \cdot \mathbf{n}| - 1)\mathbf{n} \right] \tag{B.10}$$

In the Stabilized Conservative Level Set (SCLS) approach of [8], the contour normal vector is replaced with an alternative normal vector with its magnitude diminishes away from the interface contour. The new reinitialization as proposed in [8] can be written as,

$$\frac{\partial \psi}{\partial \tau_r} = -\nabla \cdot (\psi(1-\psi)\mathbf{m}) + \nabla \cdot (\varepsilon(\nabla \psi \cdot \mathbf{m})\mathbf{m}) + \nabla \cdot ((1-|\mathbf{m}|^2)\varepsilon \nabla \psi) \tag{B.11}$$

where, $\mathbf{m} = \frac{\varepsilon \nabla \psi}{\left(\varepsilon^2 |\nabla \psi|^2 + \alpha^2 \exp(-\beta \varepsilon^2 |\nabla \psi|^2)\right)^{1/2}}$ and α and β are constants defining the regions where the vector $\mathbf{m}$ is approximately a unit vector.

All the above reinitialization equations and the additional equations are summarized along with the newly proposed reinitialization equation in Table B.4.

References

[1] E. Olsson, G. Kreiss, A conservative level set method for two phase flow, J. Comput. Phys. 210 (1) (2005) 225–246.

[2] E. Olsson, G. Kreiss, S. Zahedi, A conservative level set method for two phase flow II, J. Comput. Phys. 225 (1) (2007) 785–807.

[3] R.K. Shukla, C. Pantano, J.B. Freund, An interface capturing method for the simulation of multi-phase compressible flows, J. Comput. Phys. 229 (19) (2010) 7411–7439.

[4] J.O. McCaslin, O. Desjardins, A localized re-initialization equation for the conservative level set method, J. Comput. Phys. 262 (2014) 408–426.

[5] O. Desjardins, V. Moureau, H. Pitsch, An accurate conservative level set/ghost fluid method for simulating turbulent atomization, J. Comput. Phys. 227 (18) (2008) 8395–8416.

[6] T. Wacławczyk, On a relation between the volume of fluid, level-set and phase field interface models, Int. J. Multiph. Flow. 97 (2017) 60–77.

[7] Y. Sato, B. Nieno, A conservative local interface sharpening scheme for the constrained interpolation profile method, Internat. J. Numer. Methods Fluids 70 (4) (2012) 441–467.

[8] N. Shervani-Tabar, O.V. Vasilyev, Stabilized conservative level set method, J. Comput. Phys. 375 (2018) 1033–1044.

[9] L. Zhao, J. Mao, X. Bai, X. Liu, T. Li, J. Williams, Finite element implementation of an improved conservative level set method for two-phase flow, Comput. & Fluids 100 (2014) 138–154.

[10] T. Wacławczyk, A consistent solution of the re-initialization equation in the conservative level-set method, J. Comput. Phys. 299 (Supplement C) (2015) 487–525.

[11] R. Chiodi, O. Desjardins, A reformulation of the conservative level set reinitialization equation for accurate and robust simulation of complex multiphase flows, J. Comput. Phys. 343 (2017) 186–200.

[12] J. Brackbill, D. Kothe, C. Zemach, A continuum method for modeling surface tension, J. Comput. Phys. 100 (2) (1992) 335–354.

[13] A.J. Chorin, A numerical method for solving incompressible viscous flow problems, J. Comput. Phys. 2 (1967) 12–26.

[14] S. Parameswaran, J.C. Mandal, Development of a Conservative Level Set Method on Unstructured Meshes, in: Proceedings of 7th International Congress on Computational Mechanics and Simulation (ICCMS 2019), IIT Mandi, India, 2019, pp. 295–296.

[15] S. Parameswaran, J.C. Mandal, A novel roe solver for incompressible two-phase flow problems, J. Comput. Phys. 390 (2019) 405–424.

[16] S. Gottlieb, On high order strong stability preserving runge-kutta and multi step time discretizations, J. Sci. Comput. 25 (1) (2005) 105–128.

[17] A.L. Gaitonde, A dual-time method for two-dimensional unsteady incompressible flow calculations, Internat. J. Numer. Methods Engrg. 41 (6) (1998) 1153–1166.

[18] Y. Aiming, C. Sukun, Y. Liu, Y. Xiaoquan, An upwind finite volume method for incompressible inviscid free surface flows, Comput. & Fluids 101 (2014) 170–182.

[19] I. Tadjbakhsh, J.B. Keller, Standing surface waves of finite amplitude, J. Fluid Mech. 8 (3) (1960) 442–451.

[20] Y. Zhang, Q. Zou, D. Greaves, Numerical simulation of free-surface flow using the level-set method with global mass correction, Internat. J. Numer. Methods Fluids 63 (June 2009) (2010) 651–680.

[21] J.C. Martin, W.J. Moyce, Part IV. An experimental study of the collapse of liquid columns on a rigid horizontal plane, Philos. Trans. R. Soc. Lond. A 244 (882) (1952) 312–324.

[22] E.G. Puckett, A.S. Almgren, J.B. Bell, D.L. Marcus, W.J. Rider, A high-order projection method for tracking fluid interfaces in variable density incompressible flows, J. Comput. Phys. 130 (2) (1997) 269–282.

[23] S. Hysing, S. Turek, D. Kuzmin, N. Parolini, E. Burman, S. Ganesan, L. Tobiska, Quantitative benchmark computations of two-dimensional bubble dynamics, Internat. J. Numer. Methods Fluids 60 (11) (2009) 1259–1288.